Institutional Repository - Research Portal
Dépôt Institutionnel - Portail de la Recherche
researchportal.unamur.be
University of Namur

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Conception et implémentation de passerelles SMS - MMS - IMS

Noel, Denis

Award date:
2005

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 05. May. 2023

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année académique 2004-2005

Conception et implémentation de passerelles SMS ⟷ MMS ⟷ IMS

Denis Noel

Mémoire présenté en vue de l'obtention
du grade de Maître en Informatique.

Avant-propos

Avant tout, j'aimerais remercier certaines personnes sans qui la réalisation de ce document ainsi que le bon achèvement du stage qui en est la source n'auraient pas été possible.
Je remercie mon promoteur, Monsieur le professeur Laurent Schumacher pour ses nombreux conseils, qu'ils soient techniques ou qu'ils concernent la rédaction à proprement parler.
Je tiens ensuite à remercier Monsieur Thierry Godfroid pour l'attention qu'il m'a accordée durant mon stage au sein de l'entreprise Nextenso SA *ainsi que pour ses relectures critiques de ce document.*
Merci également à monsieur Arjun Panday pour m'avoir encadré durant mon stage. Pour ses nombreux conseils et remarques critiques qui m'ont permis de progresser.
Merci à Julien Tessot et Iann Lopez pour m'avoir fourni de la documentation précieuse qui m'a permis de mieux comprendre le monde de la télécommunication.
Merci aussi à tout le département R&D *de* Nextenso, *et tout particulièrement aux membres de l'équipe MMS, pour l'ambiance chaleureuse, détendue et constructive qui a régné pendant toute la durée de mon stage.*
Merci à Marc Gillard et Jean-François Grégoire pour avoir corrigé mon orthographe.
Merci à ma famille pour son soutien discret mais sans lequel la rédaction de ce document n'aurait pas été possible.
Il ne me reste plus qu'à souhaiter au lecteur une agréable lecture. J'espère qu'il prendra autant de plaisir à découvrir son contenu que j'en ai eu à le rédiger.

Résumé

L'évolution des technologies en matière de télécommunications ouvre la porte à de nouveaux services rendus aux utilisateurs. Pour rendre les nouveaux services compatibles avec les anciens, il est possible de mettre au point des passerelles. C'est l'objet de ce document. Après avoir introduit les prérequis nécessaires, nous allons présenter deux passerelles. Une première passerelle apppelée SMS/MMS Gateway permet à un opérateur de téléphonie mobile d'offrir à ses clients ou à des fournisseurs de contenu la possibilité de gérer des services SMS et MMS Premium. Cette passerelle a fait l'objet d'un stage au sein de la société *Nextenso S.A.* Une deuxième passerelle appelée PoC/MMS Gateway cherche à répondre aux besoins des opérateurs de téléphonie mobile en matière de diffusion des IMS. Elle permet à des utilisateurs IMS de communiquer via leur application Pusht To Talk avec des utilisateurs non-IMS par le biais du MMS.

Mots-clés : SMS, MMS, IMS, proxylet, SIP, Push To Talk

Abstract

Technologies evolution in telecommunication make possible new services. In order to make new services compatible with old ones, it's possible to conceive gateways. This is the purpose of this document. After introduce the basic knowledge, we will demonstrate two gateways. The first one is called SMS/MMS Gateway. Its purpose is to provide mobile operator with a complete solution that allows setting up Premium SMS and MMS in partnership with one or several content providers. It was the topic of an internship carried out in a company called *Nextenso S.A.* The second one is called PoC/MMS Gateway and tries to answer to the mobile operator's needs for IMS diffusion. It aims to allow IMS users to use their Push To Talk application with non-IMS users thanks to the MMS.

Keywords : SMS, MMS, IMS, proxylet, SIP, Push To Talk

Table des matières

Table des figures

Liste des abréviations

3GPP	*Third Generation Partnership Project*
3GPP2	*Third Generation Partnership Project 2*
AMR	*Adaptative Multi-Rate*
BTS	*Base Transceiver Station*
CDR	*Charging Data Record*
CSeq	*Command Sequence*
EOS	*End Of Speech*
ESME	*External Short Message Entity*
FAI	*Fournisseur d'Accès Internet*
GGSN	*Gateway GPRS Support Node*
GPRS	*General Packet Radio Service*
GSM	*Global System for Mobile*
HLR	*Home Location Register*
HTML	*Hyper Text Markup Language*
HTTP	*Hyper Text Transfer Protocol*
IETF	*Internet Engineering Task Force*
IMS	*IP Multimedia Subsystem*
LDAP	*Lightweight Directory Access Protocol*
ME	*Mobile Equipement*
MIME	*Multipurpose Internet Mail Extension*
MMS	*Multimedia Messaging Service*
MMSC	*MMS Centre*
MSC	*Mobile Switching Centre*
OMA	*Open Mobile Alliance*
P2T	*Push To Talk*
PIDF	*Presence Information Data Format*
PoC	*Push To Talk over Cellular*
RLS	*Resource List Server*
RNC	*Radio Network Controllers*
RPID	*Rich Presence Information Data Format*
RTCP	*RTP Control Protocol*
RTP	*Real-time Transport Protocol*
SDP	*Session Description Protocol*
SGSN	*Serving GPRS Support Node*
SIM	*Subscriber Identity Module*
SIP	*Session Initial Protocol*
SME	*Short Message Entity*
SMS	*Short Message Service*
SMSC	*SMS Centre*

SOAP	*Simple Object Access Protocol*
SOS	*Start Of Speech*
TCP	*Transmission Control Protocol*
TDMA	*Time Division Multiple Access*
UAC	*User Agent Client*
UAS	*User Agent Server*
UDP	*User Datagram Protocol*
UMTS	*Universal Mobile Telecommunication System*
URI	*Universal Resource Identifier*
URL	*Universal Resource Locator*
UTRAN	*Universal Terrestrial Radio Access Network*
VAS	*Value Added Service*
VLR	*Visitor Location Register*
VoIP	*Voice over IP*
WAP	*Wireless Application Protocol*
XML	*eXtend Markup Language*
XCAP	*XML Configuration Access Protocol*

Introduction

La téléphonie mobile a connu un essor considérable ces dix dernières années. Rares sont les personnes qui à l'heure actuelle ne disposent pas d'un téléphone portable. Les technologies des réseaux mobiles ainsi que celles des terminaux sont en constante évolution. Cette évolution va de pair avec une évolution des services rendus aux utilisateurs.

Le SMS fait désormais partie de notre quotidien, il est devenu un moyen de communication presque incontournable.

L'arrivée du MMS et la possibilité d'envoyer du contenu multimédia montrent une évolution des services qui ne se limitent plus à une communication de "*personne à personne*" mais à un "*échange de contenu*".

Avec l'émergence des IMS, la téléphonie mobile de demain fera converger les différents réseaux mobiles avec l'Internet, permettant ainsi la mise au point de nouveaux types de services. Les bonnes vieilles conversations téléphoniques ne seront qu'une des nombreuses possibilités offertes par les terminaux mobiles.

Pour permettre une telle évolution, il est indispensable de rendre les nouveaux services compatibles avec les anciens[1]. Une des possibilités pour permettre de telles compatibilités est la mise au point de passerelles capables de rendre interopérables les services de générations différentes.

Ce document est consacré à l'étude de ce genre de passerelle.

Après avoir présenté les évolutions en matière de technologies et en matière de services dans la téléphonie mobile, nous présenterons dans la première partie les différentes technologies nécessaires à la compréhension des passerelles décrites dans les deux dernières parties.

Dans la deuxième partie, nous présenterons en détail une passerelle que nous avons implémentée dans le cadre d'un stage au sein de la société *Nextenso S.A* : la SMS/MMS Gateway.

Enfin, dans de la troisième partie, nous proposerons et analyserons une passerelle originale qui a pour but de faire interagir des utilisateurs IMS avec des utilisateurs non-IMS :

[1] Appelés également services *legacy*.

la PoC/MMS Gateway.

0.1 Evolution des technologies dans la téléphonie mobile

Cet historique est inspiré de [3], [4] et [14].

0.1.1 Le GSM

Avant l'introduction du *Global System for Mobile*, plus connu sous le nom de GSM, les réseaux mobiles implémentés dans différents pays étaient incompatibles. La norme GSM est née d'une volonté de mettre fin à la cacophonie qui régnait alors en matière de réseau de radiotéléphone.

Un réseau GSM est un réseau de télécommunication qui offre des services de transmission de données en mode circuit. Il fonctionne selon un dispositif permettant de grouper plusieurs communications sur une même liaison que l'on appelle TDMA. Cela signifie que plusieurs mobiles peuvent utiliser la même fréquence d'émission et la même fréquence de réception. Pour qu'ils ne se perturbent pas, cette fréquence leur sera attribuée à tour de rôle.

Dans les systèmes de communication sans fil, la bande radio représente une ressource rare, qu'il faut utiliser le plus efficacement possible. C'est dans un tel dessein qu'a été mis au point le concept de cellule sur lequel se base le réseau GSM. Avec ce concept, une même ressource radio (caractérisée par une bande de fréquence et un intervalle de temps) peut être utilisée par plusieurs utilisateurs sans interférence signigicative s'ils sont séparés par une distance minimum. En fonction de la topographie, en ville ou à la campagne, chaque cellule peut avoir un diamètre d'activité de quelques centaines de mètres à quelques kilomètres. Cela est directement lié à la densité de population. En effet, la densité de population sera beaucoup plus élevée dans un milieu urbain que dans un milieu rural. Partant du principe qu'il n'existe plus d'interférences entre les cellules utilisant la même fréquence lorsque celles-ci se trouvent séparées par quelques cellules, il sera possible de réutiliser les mêmes canaux (fréquences) de nombreuses fois.

L'architecture d'un réseau GSM est présentée à la figure A.1. Voici une succincte description de ses principaux éléments :

La station mobile Elle correspond au téléphone sans fil. Il est composé d'un équipement mobile (ME) qui s'occupe des transmissions radio et d'un SIM (Subscriber Identity Mobile) matérialisé par une carte à puce, qui permet d'identifier un utilisateur abonné sur le réseau.

Le sous-système de station de base Il est composé d'une ou plusieurs BTS (Base Transceiver Station) qui sert d'interface avec toutes les stations mobiles présentes dans la

cellule qu'elle contrôle. Les BTS sont connectées à une BSC (Base Station Controller) qui gère leur coordination.

Le sous-système réseau Il comprend trois parties :

1. **Le MSC** (Mobile Switching Centre) gère l'interconnexion du réseau GSM avec le réseau téléphonique public. Il génère toutes les informations de facturation des utilisateurs et gère la complexité des connexions dues aux déplacements réalisés pendant la communication.
2. **Le HLR** (Home Location Register) est la base de données centrale contenant toutes les informations administratives relatives aux abonnés du réseau, ainsi que la zone de service où le mobile de cet abonné s'est connecté ou a été signalé la dernière fois.
3. **Le VLR** (Visitor Location Register) est une base de données qui contient des informations dynamiques, concernant les utilisateurs connectés au réseau mobile, telles que la localisation géographique.

0.1.2 Le GPRS

GPRS est l'acronyme de *General Packet Radio Service*, il s'agit d'une extension de la norme GSM. La transmission de données par commutation de circuits, telle que celle du réseau GSM, ne représente pas une façon efficace d'exploiter les ressources radios. Le GPRS répond à ce problème en permettant la transmission de données par commutation de paquets.

Avec un tel système, les données transmises sont découpées en blocs de petite taille appelés paquets. Chaque paquet se voit attribuer en-têtes et informations de contrôle. Les ressources radio sont de cette façon utilisées beaucoup plus efficacement puisque un seul canal transporte simultanément plusieurs communications en parallèle. La conséquence directe est que le GPRS offre une bande passante plus élevée (jusqu'à 171,2 kb/s contre 57,6 kb/s pour le GSM).

La figure A.2 nous montre l'architecture globale d'un réseau GPRS. Deux composants réseaux ont été intégrés au sous-système réseau :

1. **Le SGSN** (Serving GPRS Support Node) agit comme un routeur de paquets pour toutes les stations mobiles présentes dans une région géographique donnée.
2. **Le GGSN** (Gateway GPRS Support Node) assure l'interface entre le *Core Network* et l'Internet.

0.1.3 L'UMTS

UMTS (*Universal Mobile Telecommunication System*) désigne une technologie d'accès radio dite de $3^{\text{ème}}$ génération qui succède au GSM et au GPRS mais en utilisant de nou-

velles fréquences (dans la bande des 1900 - 2100 MHz). Cette norme permet d'offrir de la téléphonie mobile classique ainsi que du transport de données (en mode commutation de paquets) à un débit théorique de 2 Mb/s. Un mobile UMTS est à même de supporter un accès Internet haut débit. L'UMTS doit se développer en deux phases. Durant la première phase, l'architecture doit être suffisamment sophistiquée pour offrir des services tels que le trafic multimédia mixte ou le trafic de voix en temps réel.

Une telle architecture est présentée à la figure A.3. Le sous-système de station de base est remplacé par l'UTRAN (*Universal Terrestrial Radio Access Network*) qui est composé de Noeuds B, responsables de la transmission d'informations vers et en provenance des terminaux, pour une ou plusieurs cellules. Les Noeuds B sont interconnectés avec les RNC (*Radio Network Controllers*) qui contrôlent les ressources dans le système et servent d'interfaces avec le coeur réseau.

Alors que la première phase permet à un opérateur de réseau mobile de convertir son réseau en UMTS en se basant sur les réseaux GSM et GPRS existants, la seconde phase aura pour but de se défaire des reliquats liés au GSM en offrant une architecture uniquement basée sur une transmission de données par commutation de paquets.

Le *Core Network* de la seconde phase correspondra aux *IP multimedia Subsystems* (IMS) qui permettront des services tels que le Voice Over IP (VoIP). Nous reviendrons plus en détails sur les IMS lors de la description des services liés à la téléphonie mobile.

0.2 Evolution des services dans la téléphonie mobile

0.2.1 SMS

Le *Short Message Service* (SMS) est un service sans fil qui permet la transmission de messages alphanumériques de taille limitée (160 caractères) entre deux terminaux.

Nous allons brièvement décrire l'architecture SMS qui est illustrée à la figure A.4 :

Le SME (*Short Message Entity*) Il s'agit de l'élément qui envoie ou reçoit un message. Un SME peut être une application embarquée dans un terminal, mais il peut s'agir d'un serveur Internet distant, un télex, etc ... Un SME peut être également un serveur qui s'interconnecte au SMSC directement ou par l'intermédiaire d'une passerelle. Un tel SME est connu sous le nom de ESME (*External SME*). Par la suite, nous appellerons OSME, le SME à l'origine d'un message et DSME, le SME à qui le message est destiné.

Le SMSC (*SMS Center*) Il joue un rôle clé dans l'architecture. Il s'occupe de router les messages entre deux SME. Il s'occupe également de ce que l'on appelle le *Store and Forward* : lorsqu'un DSME n'est pas disponible (c'est le cas lorsque le terminal dans lequel il se trouve n'est pas connecté au réseau), le SMSC se charge de stocker les messages qui lui sont destinés jusqu'au moment où le DSME devient disponible.

Lorsque celui-ci se connectera au réseau, le HLR enverra une notification au SMSC qui pourra lui délivrer le message. Un OSME peut également définir une période de validité au-delà de laquelle le message sera supprimé s'il n'a pas été récupéré par le DSME.

La passerelle Email (*Email Gateway*) Elle rend le SMS et l'e-mail interopérable en connectant le SMSC à l'Internet.

Les deux principales caractéristiques d'un SMS sont l'envoi et la réception d'un message :

1. L'envoi d'un message, aussi connu sous le nom de SM-MO (*Short Message-Mobile Originated*) correspond aux messages qui proviennent d'un OSME à destination du SMSC.
2. La réception d'un message, aussi connu sous le nom de SM-MT (*Short Message-Mobile Terminated*) correspond aux messages qui proviennent d'un SMSC à destination d'un DSME.

Il est également possible pour un OSME de demander un rapport de statut. Ce rapport indique si le message d'origine a été ou non délivré à son destinataire.

Des accords commerciaux entre opérateurs mobiles permettent l'échange de SMS entre utilisateurs n'appartenant pas au même réseau.

0.2.2 MMS

Le MMS (*Multimedia Messaging Service*) est un service qui permet à plusieurs utilisateurs de téléphones portables de s'échanger des messages multimédias. Notons que dans la suite, par abus de langage lorsque nous parlerons d'un MMS, nous parlerons aussi bien du service que du message lui-même.

L'évolution des technologies dans les réseaux de téléphonie mobile, la nécessité de faire converger le marché du mobile avec les services existant sur Internet ainsi que les demandes de plus en plus exigeantes de la clientèle, sont à l'origine de l'introduction du MMS.

Les besoins des utilisateurs ont évolué, les applications centrées sur la voix ne sont plus suffisantes, il faut désormais se diriger vers des services de visualisation de contenu. Le MMS répond à ces demandes. Il n'est plus limité à 160 caractères comme le SMS, et a de plus la possibilité d'être enrichi d'un contenu multimédia (image, son, vidéo).

Avec le MMS, un message peut être structuré comme une présentation de transparents à l'instar de ce que propose le logiciel *Microsoft PowerPoint*. Chaque transparent est composé d'éléments multimédias et est présenté durant une certaine période de temps qui peut être configurée par l'utilisateur à l'origine du message.

Contrairement au SMS, le MMS permet d'envoyer un message à plusieurs destinataires. Alors que le SMS permettait juste de savoir si un destinataire avait bien reçu son message,

le MMS permet en plus de savoir si l'utilisateur l'a lu. Un MMS peut également avoir différentes priorités et classes. On retrouve là une grande similitude avec l'e-mail.

Un nouveau concept est celui de la notification qui permet un retrait immédiat ou différé du message. L'envoi et la réception d'un MMS sont donc basés sur un système asynchrone. Lorsqu'un utilisateur reçoit un message, il est notifié. Cette notification est effectuée par SMS, si l'utilisateur n'est pas connecté à l'Internet, ou directement via la connexion s'il est connecté. Quelque soit le mode de réception, il est transparent pour l'utilisateur final. Une fois la notification reçue, il peut télécharger le message afin de le consulter.

Le concept de validité de message est également présent. Ainsi si notre utilisateur ne télécharge pas son message, au bout d'un certain temps il sera effacé du serveur de l'opérateur.

L'hétérogénéité des terminaux fait que leurs capacités sont différentes. Certains téléphones ont une capacité d'affichage moins importante que d'autres et ne peuvent par conséquent pas afficher correctement tout type de contenu. A cette fin, il est possible de procéder à une adaptation de contenu pour que celui-ci soit en adéquation avec les capacités du terminal auquel il est destiné. Les constructeurs de terminaux mettent à disposition, au moyen d'une URL publique, les caractéristiques techniques qui sont résumées dans un format standardisé appelé *UAProf* grâce auquel il est possible de connaître le type d'adaptation nécessaire. Dans le cas extrême, lorsqu'un terminal ne supporte pas le MMS, un SMS est envoyé au destinataire l'invitant à se connecter à une adresse Internet particulière afin de consulter son message.

Les terminaux sont également limités au niveau de leur capacité de stockage. Pour répondre à ce problème, les concepteurs du MMS ont mis au point le concept de stockage permanent. Un utilisateur peut disposer d'une boîte de message (*MMBox*) afin de conserver ses messages sans surcharger son téléphone.

Après avoir donné les principales caractéristiques du MMS, nous allons présenter son architecture.

Architecture

Le MMS fait interagir différents types de réseaux : avec ou sans fil, Internet, ... L'un des objectifs du MMS est, comme nous l'avons signalé plus haut, de fournir un service interopérable avec les systèmes de messagerie mobile tels que le SMS mais aussi avec des systèmes de messagerie fixe tels que l'e-mail.

Un environnement MMS (MMSE) correspond à l'ensemble des éléments MMS sous le contrôle d'un même administrateur (le fournisseur de MMS) qui est en charge de fournir le service MMS à tous ses abonnés. Des utilisateurs appartenant à des environnements MMS différents ont également la possibilité de s'échanger des messages comme nous le verrons

plus loin.

La figure A.5 nous montre l'architecture MMS. Les éléments clés de cette architecture sont le *MMS Relay* et le *MMS Server*. Le *MMS Relay* a la charge de router les messages au sein d'un environnement MMS ainsi qu'en dehors de celui-ci. Le *MMS Server* se charge de stocker les messages en attente. Ensemble, le *MMS Relay* et le *MMS Server* forment ce que l'on appelle le MMSC. Le MMSC est au MMS ce que le SMSC est au SMS.

En plus du routage et du stockage des messages, le MMSC est chargé de l'adaptation de contenu des différents messages. Pour ce faire, il doit récupérer les informations concernant les capacités du terminal contenues dans l'*UAProf*. Pour faciliter l'adaptation de contenu, les terminaux sont classés par groupe en fonction desquels on peut appliquer des transformations pour le transfert de MMS d'un groupe à l'autre.

Le MMSC est également chargé de générer des CDR (Charging Data Records). Ceux-ci sont utilisés par les opérateurs pour facturer les utilisateurs pour leur utilisation des ressources du réseau.

Comme nous pouvons le voir sur la figure A.5 l'environnement du MMS est très hétérogène. Pour permettre une telle hétérogénéité, plusieurs interfaces ont été définies :

MM1 C'est l'interface clé du MMSC, c'est par celle-ci qu'un terminal communique avec le MMSC.

MM2 Interface par laquelle communiquent les deux composants du MMSC : le *MMS Relay* et le *MMS Server*. La plupart des solutions commerciales offrent les deux composants dans le même produit. Cette interface est donc propriétaire et non standardisée.

MM3 Cette interface est utilisée par le MMSC pour communiquer avec les serveurs externes. Elle permet par exemple la compatibilité entre le MMS et l'e-mail.

MM4 Il s'agit de l'interface par laquelle communiquent deux MMSC. C'est grâce à cette interface que deux utilisateurs appartenant à deux environnements différents peuvent interagir.

MM5 Elle permet les interactions avec les différents éléments du réseau. C'est par exemple par cette interface que le MMSC interroge le HLR afin d'obtenir des informations de routage du message.

MM6 Elle permet les interactions avec le serveur LDAP qui contient la base de données des utilisateurs.

MM7 Elle permet les interactions avec les applications VAS (*Value Added Service*). Nous expliquerons cette interface plus en détail dans le point suivant.

MM8 C'est par cette interface que s'effectuent les opérations de facturation des clients.

L'interface MM7

Les anciens services de messagerie mobile, tels le SMS, ne supportent pas de contenu multimédia. Ces services conviennent pour des scénarii dit : "*de personne-à-personne*", mais ils sont limités pour les scénarii dit "*de contenu-à-personne*". Grâce au MMS, les fournisseurs VAS peuvent générer du contenu multimédia. Les applications VAS fournissent un service ou un contenu à leurs abonnés, comme par exemple la gestion d'un album photo par MMS, ou une liste de diffusion via laquelle il est possible de recevoir chaque jour les informations boursières. L'interface MM7 est dédiée à la communication entre VAS et MMSC. Cette interface est basée sur le protocole SOAP qui est transporté sur le protocole HTTP.

Dans cette configuration, un serveur VAS peut :

- soumettre un message à un MMSC afin qu'il le délivre à une ou plusieurs personnes ;
- annuler l'envoi d'un message ;
- remplacer un message qui n'a pas encore été délivré ;
- recevoir des messages de la part de terminaux via l'interface MM1.

Dans un tel contexte, il est évident qu'un fournisseur VAS aura au préalable dû conclure des accords commerciaux avec le fournisseur de MMS.

Format d'un MMS

Un MMS peut prendre différentes formes afin de pouvoir être transporté sur différents types de bande passante. Lorsqu'il est transféré sur des protocoles Internet, entre un MMSC et un serveur VAS par exemple, il est encodé au format texte. Par contre, lorsqu'il est transféré entre un terminal et un MMSC, la bande passante étant plus limitée, le message sera encodé en format binaire pour pouvoir être transféré de manière plus efficace.

Le format d'un MMS est dérivé du format MIME, il est composé d'une enveloppe et d'un contenu. Les enveloppes contiennent les données relatives au routage du message :

- l'adresse de l'expéditeur (*From*) ;
- l'adresse du (des) destinataire(s). Les adresses sont organisées en adresses primaires et adresses secondaires. Les adresses primaires correspondent au champ *To* et les adresses secondaires correspondent aux champs *Cc* et *Bc*.

ainsi que les caractéristiques du message :

- classe ;
- priorité ;
- sujet ;
- ...

Le contenu est consitué de différentes parties qui peuvent encapsuler texte, image, son ou vidéo. Chaque partie est composée d'informations sur l'objet média encapsulé et de l'objet

média lui même.

0.2.3 IMS

L'émergence de l'Internet et le succès de l'e-mail ont créé un nouveau moyen de communication qui a été propice à de nouveaux médias. Le succès de la téléphonie mobile et les nouveaux services tels que le SMS, le MMS combinés au développement des réseaux sans fil ont apporté de nouveaux défis en matière de moyen de communication. La bonne vieille téléphonie est aujourd'hui sur le point d'être complétée par une nouvelle série de services dit de communication de personne à personne et de partage de contenu. Il est crucial de comprendre qu'il devient extrêmement facile pour n'importe qui de produire du contenu numérique. Il est donc crucial de fournir la possibilité aux utilisateurs finaux de communiquer avec leurs amis et d'échanger ce contenu.

Les terminaux mobiles deviennent de plus en plus sophistiqués, ils sont en train de se muter en véritables ordinateurs de poche, permettant aux utilisateurs finaux de vérifier leurs mails, surfer sur Internet, installer des programmes, etc ...

Cette dimension est à la fois propice et dangereuse pour les opérateurs de réseaux mobiles :

- propice car des terminaux performants leur permettent d'offrir des services plus performants et plus diversifiés ;
- dangereuse car les nouvelles technologies permettent à des tierces parties d'offrir des services aux utilisateurs d'un opérateur de réseau mobile sans que celui ne bénéficie de revenu lié au service lui-même. Les fournisseurs de services traditionnels de l'Internet tel que *MSN Messenger* de Microsoft fournissent leur service sur des terminaux mobiles. *Skype*, qui est un logiciel de téléphonie qui utilise la technologie Voice Over IP offre des prix qui défient la concurrence.

Les opérateurs risquent de devenir de simples transporteurs de paquets ne tirant de bénéfices que sur le transport des données et non sur la valeur ajoutée des services offerts. L'un des défis majeurs des opérateurs de réseaux mobiles est de trouver une solution offrant des services à leurs utilisateurs dans un environnement dont ils ont le contrôle.

Selon [14], l'OMA a reconnu qu'il n'était pas bénéfique pour chaque fournisseur de service d'avoir son propre mécanisme de sécurité, de tarification, de gestion de session, etc ... Au contraire, les fournisseurs de services devraient pouvoir tirer parti d'une infrastructure qui leur offre ces possibilités.

Un environnement IMS offre aux fournisseurs de services ce genre de possibilités. Pour rappel, IMS est l'acronyme de *IP Multimedia Subsystem*, il s'agit d'un environnement dont l'architecture standardisée offre des services de communication basés sur le protocole IP aux opérateurs de réseaux mobiles.

L'architecture IMS est composée de deux parties distinctes :

1. Le *Core Network* qui est spécifié par les organismes de standardisation 3GPP/3GPP2. Cette partie des IMS sort du cadre de ce document, nous n'entrerons par conséquent pas dans les détails.
2. La partie applicative qui contient différents types de services qui sont spécifiés par l'OMA.

Avec un environnement IMS, un opérateur peut offrir des services multimédias incluant le VoIP, la diffusion d'images et de vidéos, la vidéoconférence, etc ... Ce genre de services sont destinés à des terminaux dit de $3^{\text{ème}}$ génération (3G) mais sont également accessibles pour des terminaux 2,5 G. Les terminaux fixes tels que des ordinateurs personnels pourront également profiter de ce genre de services s'ils bénéficient d'une connexion à l'Internet.

Dans [2], la partie applicative est scindée en deux, les "*enablers*" et les services.

Les "*enablers*"

Les "*enablers*" sont des composants qui offrent une interface standard à des services à valeur ajoutée internes et externes. Ils sont définis par le 3GPP et l'OMA.

Ils gèrent notamment l'identification et l'autorisation. Il faut bien distinguer ces deux concepts :

1. L'identification est le processus qui consiste à vérifier si un utilisateur est bien la personne qu'il prétend être. Cette vérification se fait à l'aide d'un nom d'utilisateur et d'un mot de passe.
2. L'autorisation est le processus qui consiste à vérifier si la personne identifiée a effectivement la permission de bénéficier du service dont elle est demandeuse.

Grâce à cet environnement, il devient plus facile pour une tierce partie de développer une application car la partie "gestion des utilisateurs " est prise en charge par l'infrastructure IMS. De plus l'opérateur garde le contrôle de son environnement, il n'est plus un simple transporteur de paquets. La figure 1 illustre ce propos.

Les services

Il existe différents services multimédias à valeur ajoutée tels que

- le Push To Talk : Service de messagerie vocale instantanée que nous expliquerons plus en détail dans la suite de ce document ;
- l'Instant Messaging : Service de messagerie instantanée de type *MSN Messenger* ;
- le Call Back Service : Service permettant à un utilisateur d'être averti lorsqu'un autre utilisateur se connecte ;
- ...

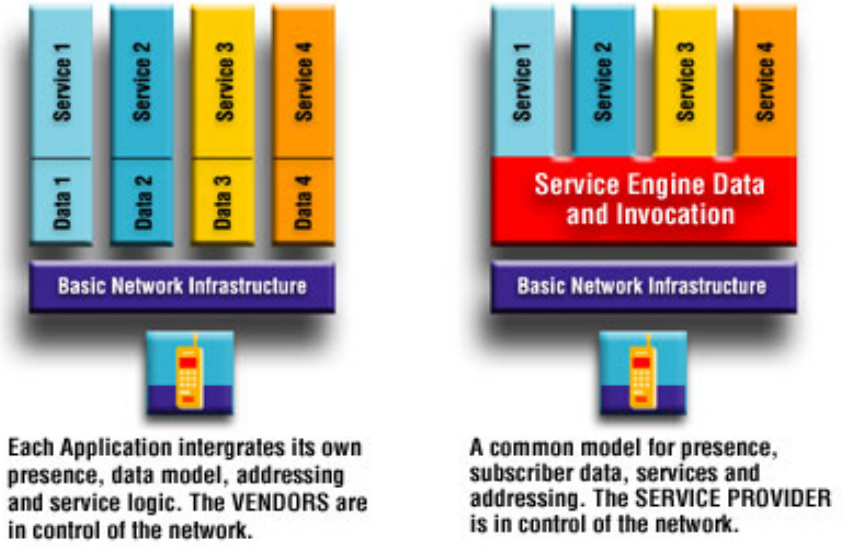


FIG. 1 – Illustration tirée de [1].

Après avoir dressé un bref historique des technologies et services liés à la télécommunication sans fil, nous allons, dans la partie suivante, présenter les outils nécessaires à la compréhension des passerelles décrites en deuxième et troisième partie, c'est-à-dire le protocole SIP, l'application Push To Talk et la *Proxy Platform*.

Première partie

Prérequis

Chapitre 1

SIP

Ce chapitre a pour but de présenter succinctement le protocole SIP. Il est inspiré de [2, Ch 8] et [16]. Le lecteur désireux d'approfondir le sujet pourra consulter également [8, 7, 15].

SIP (Session Initial Protocol) est un protocole réseau de la couche application conçu pour établir, modifier et terminer des sessions multimédias entre plusieurs utilisateurs sur un réseau IP. Outre la gestion de sessions, il met à disposition plusieurs services tels que la mise en attente, les transferts et les déviations d'appels. Il introduit également la notion de "*personal mobility*" soit différents terminaux pour une même personne, par exemple un téléphone fixe et un téléphone portable. Le principe du protocole SIP est d'ajouter un niveau d'indirection dans l'identification d'une personne. La référence ne se fait pas directement par rapport à l'adresse du terminal de la personne mais par rapport à une SIP URI qui est associée à celle-ci.

Une SIP URI suit le même format qu'une adresse e-mail : `sip:userinfo@hostport[parameters][headers]`

Userinfo Il s'agit d'un nom d'utilisateur ou d'un numéro de téléphone.

Hostport Le nom de domaine ou l'adresse numérique réseau et le port.

Parameters Paramètres spécifiques de l'URI (la couche transport, le *time to live*, ...).

Headers D'autres informations rarement utilisées.

Voici quelques exemples de SIP URI :

- `sip:denis.noel@info.fundp.ac.be` ;
- `sip:denis@info.fundp.ac.be;transport=tcp` ;
- `sip:+32476875597@proximus.be;user=phone` ;
- `denis@138.48.32.100:8001`.

SIP est indépendant de la couche transport, il fonctionne aussi bien sur TCP que sur UDP. Cependant, en pratique, le protocole UDP est préféré car il est très rapide et sans

connexion.

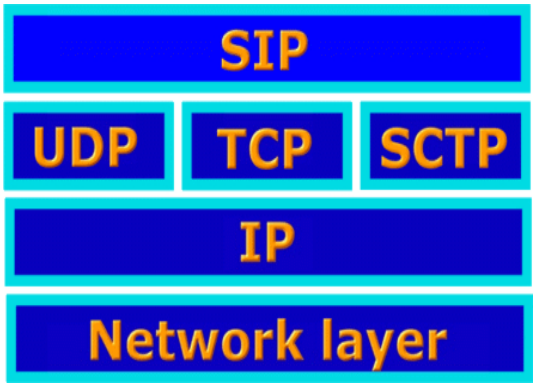


FIG. 1.1 – SIP est indépendant de la couche transport.

SIP est un protocole basé sur des requêtes et des réponses. Celles-ci sont faites d'une partie en-tête et d'une partie corps, à l'instar du protocole HTTP par exemple. Il faut bien comprendre que SIP ne transporte pas de contenu multimédia tel que du son mais bien des messages textuels dont le seul but est d'établir une liaison entre deux ou plusieurs agents. SIP ne permet pas non plus de donner la description d'une session. Celle-ci est faite à l'aide d'un document se trouvant dans le corps des messages SIP et exprimé dans un autre protocole. SIP tout seul n'est donc pas utile, il n'est qu'un composant d'une architecture média complète qui inclut des protocoles tels que :

- SDP (Session Description Protocol) qui est le protocole permettant de donner la description d'une session multimédia (type de média, type de compression, ports, ...) ;
- RTP (Real-Time Transport Protocol) qui est un protocole permettant de transporter des données audios ou vidéos.

Des systèmes tels que le VoIP respectent une telle architecture.

1.1 Les messages SIP

Les messages SIP sont composés d'une en-tête obligatoire et d'un corps facultatif. C'est le corps des messages qui contient les documents SDP nécessaires à l'établissement d'une connexion.

L'en-tête consiste en une série de lignes, appelées champs, qui peuvent être parcourues facilement. Elle contient diverses informations nécessaires au routage du message à travers les serveurs SIP. Un message peut être soit une requête émanant du client, soit une réponse provenant d'un serveur.

La figure 1.2 nous montre le format d'un message SIP.

L'en-tête est composée de champs obligatoires :

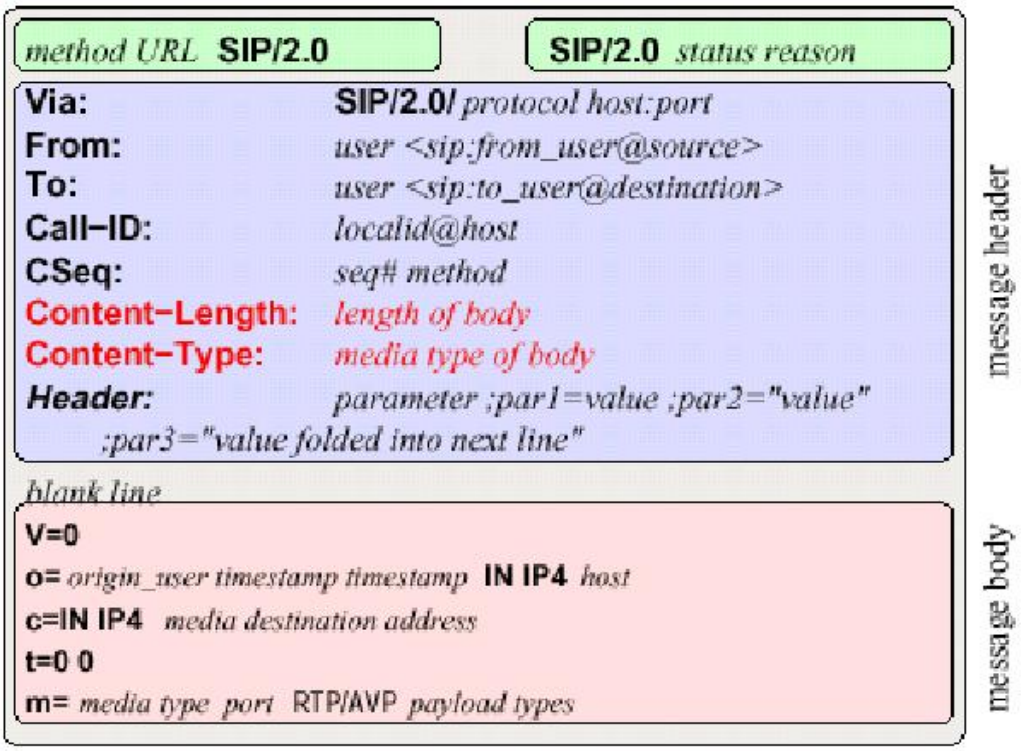


FIG. 1.2 – format d'un message SIP tiré de [16].

1. **method** contient la méthode aussi appelée requête. Il existe plusieurs types de méthode, nous les décrirons plus loin dans ce chapitre.
2. **Via** donne la liste des serveurs par lesquels le message a transité. Ce champ est utilisé afin de détecter les boucles.
3. **From** contient l'émetteur du message.
4. **To** contient le destinataire final.
5. **Call-ID** est un identifiant unique dans le temps et dans l'espace.
6. **Cseq** (Command Sequence) est un numéro de séquence utile à la détection de doublons lors du renvoi d'un même message.

Le corps de certains messages peut contenir un document SDP

1.1.1 Les requêtes SIP

SIP est composé de 6 requêtes de base :

1. *REGISTER* permet à un utilisateur de se faire connaître sur un réseau SIP en notifiant à ce réseau l'URI par laquelle il souhaite se faire appeler ainsi que l'adresse IP correspondant à cet URI.
2. *INVITE* est utilisée pour établir une session média entre deux utilisateurs. Le corps du message contient de l'information sur le média de l'appelant (sous forme d'un

document SDP). Une requête *INVITE* crée un Call-ID utilisé pour la durée de l'appel. Il est possible de renvoyer un *INVITE* durant une session existante (re-*INVITE*), ceci est utile pour la modification des paramètres d'une session. Le champ Cseq est incrémenté pour chaque nouvelle requête contenant le même Call-ID. C'est ce numéro qui permet de distinguer un re-*INVITE* d'un message *INVITE* original.

3. *ACK* (acknowledge) utilisé uniquement pour confirmer la demande de liaison d'une requête *INVITE*. Le corps du message peut également contenir la description de la session sous forme d'un document SDP, si ces informations n'étaient pas présentes dans le corps du message *INVITE* initial.

4. *BYE* est utilisée par l'agent désirant mettre un terme à une communication. A la suite de ce message, tout échange média est interrompu.

5. *CANCEL* met fin à une conversation en cours d'établissement (ce message est donc utilisé entre un *INVITE* et un *ACK*).

6. *OPTION* permet d'interroger un agent sur ses caractéristiques techniques. L'agent répondra à ce message comme s'il s'agissait d'un message *INVITE*.

Le protocole SIP est en constante évolution et d'autres types de requête sont venus s'ajouter, notamment pour la gestion des notifications des événements. Un utilisateur peut souscrire à l'événement d'un autre utilisateur et recevoir la notification de l'état de cet événement ainsi que de tout changement d'état de celui-ci. Deux requêtes sont utiles pour la gestion de notification des événements :

1. *SUBSCRIBE* est la méthode utilisée pour souscrire aux événements d'un utilisateur.

2. *NOTIFY* est la méthode utilisée pour recevoir les notifications de l'état d'un événement.

Nous venons de montrer comment il est possible de souscrire à un événement et comment obtenir la notification de l'état de celui-ci. Ceci dit, rien de tout cela n'est possible si les utilisateurs n'ont pas la possibilité de publier l'état de leurs événements. Cette possibilité est offerte par la requête *PUBLISH*.

Nous reviendrons plus en détails sur ces requêtes dans le chapitre 9

1.1.2 Les réponses SIP

Les réponses ont été créées sur le modèle des réponses HTTP. Elles sont divisées en 6 classes. Le lecteur pourra les trouver à l'annexe B.

1.2 Les composants SIP

Le protocole SIP n'est pas limité à la définition des types de message, leur format, etc ... Il spécifie également les différents éléments qui constituent son réseau ainsi que leur fonctionnement :

Les terminaux Il s'agit soit de téléphone SIP ou de PC équipé d'un logiciel adéquat, d'une carte son et d'un micro. Ils sont capables d'émettre et de recevoir des messages SIP. Un terminal (également appelé agent) est à la fois client et serveur. Client puisqu'il émet des requêtes vers d'autres terminaux ; dans ce cas, le terminal qui répondra à sa requête agira en tant que serveur. Dans la suite de ce document, nous appellerons UAC (User Agent Client) un terminal agissant en tant que client et UAS (User Agent Server) un terminal agissant en tant que serveur.

Les serveurs d'enregistrement Ils sont utilisés par les services de localisation. C'est à eux que sont destinées les requêtes *REGISTER*. Ils permettent à un terminal d'annoncer sa localisation.

Les serveurs de localisation Ces serveurs, basés sur une base de données, un serveur LDAP ou un simple fichier texte, permettent de mémoriser les différents utilisateurs, leurs droits, leurs mots de passe etc... ainsi que leur position actuelle.

Les serveurs de redirection Ils sont utilisés pour rediriger les appels vers un agent, l'adresse SIP ne donnant aucun renseignement sur la localisation de celui-ci. Les serveurs de redirection reçoivent les messages *INVITE* émanant d'un terminal appelant, recherchent la position du terminal appelé par l'appelant auprès d'un serveur de localisation et renvoient la localisation à l'émetteur du message *INVITE*.

Les proxies Ils remplissent la même fonction que les serveurs de redirection, mais sont plus transparents. Le terminal appelant dialogue avec un proxy comme s'il s'adressait directement au terminal appelé. Le proxy se charge de retransmettre les messages à celui-ci.

Après avoir énuméré les différents éléments constituant un réseau SIP, nous allons présenter leurs interactions. Nous montrerons un exemple expliquant la façon dont un terminal annonce sa localisation, un exemple expliquant l'établissement d'une session, et enfin, un exemple expliquant la fermeture d'une session.

1.3 Annonce de sa localisation

La figure 1.3 nous montre le diagramme de séquence expliquant comment un terminal enregistre sa localisation. Il envoie un message SIP *REGISTER* au serveur d'enregistrement. Celui-ci annonce au serveur de localisation l'adresse IP à associer à la SIP URI.

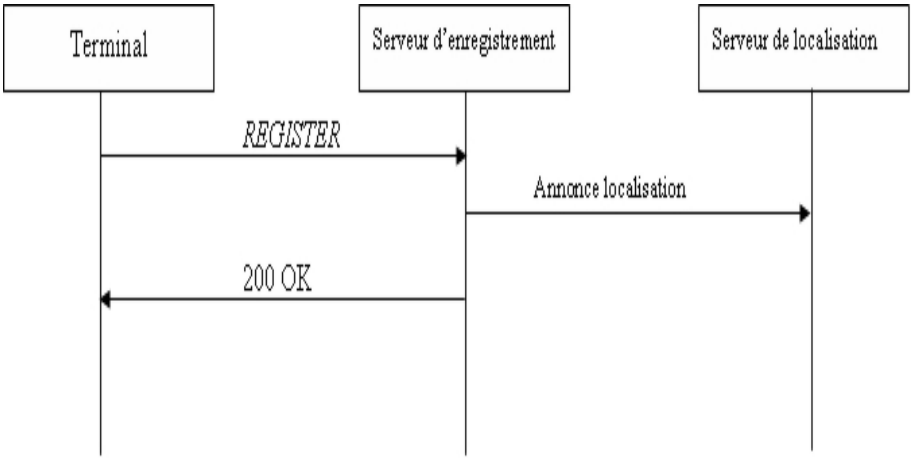


FIG. 1.3 – Annonce de la localisation.

1.4 Etablissement d'une session

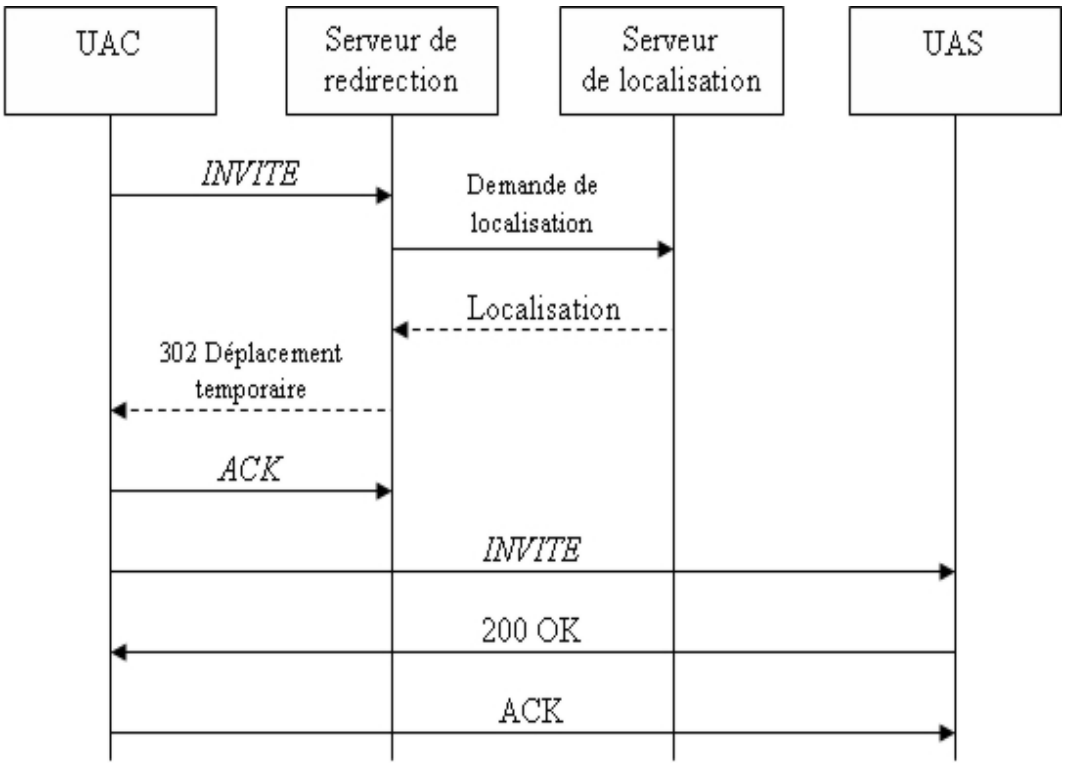


FIG. 1.4 – Etablissement d'une session.

1. Pour commencer l'UAC émet un message *INVITE* en direction du serveur de redirection, celui-ci contient la SIP URI de l'UAS.
2. Le serveur fait une demande de position au serveur de localisation.
3. Le serveur obtient la position.
4. Le serveur émet un message de déplacement temporaire (302) en direction de l'UAC.

5. L'UAC confirme par un *ACK* au serveur de redirection.
6. L'UAC émet un nouveau message *INVITE* avec la nouvelle adresse obtenue.
7. L'UAS reçoit la requête *INVITE* et émet un 200 OK.
8. L'UAC confirme la réception du 200 OK par un ACK.

1.5 Fermeture d'une session

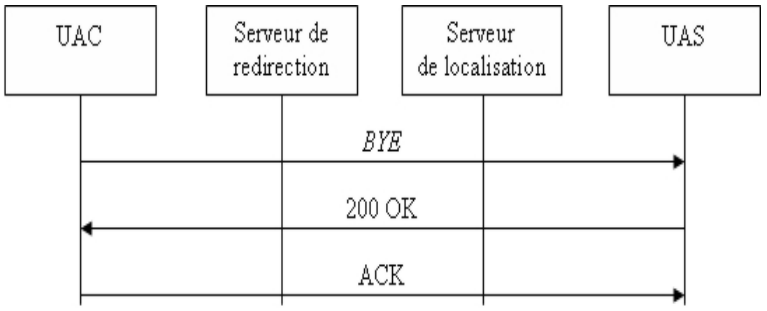


FIG. 1.5 – Fermeture d'une session.

Le scénario est illustré à la figure 1.5. Lors de la fermeture de la session, l'UAC connaît l'adresse IP de l'UAS. Dès lors, il ne lui est plus nécessaire d'interroger le serveur de redirection.

Chapitre 2

Le Push To Talk

Ce chapitre est inspiré de [12].

Le Push To Talk (P2T) est un service vocal similaire à la téléphonie classique, mais avec des caractéristiques spécifiques. Le principal aspect de ce service est le fait qu'à n'importe quel moment d'une conversation, le flux vocal est unidirectionnel.

La communication est de type "*Talkie-Walkie*", avec l'acquisition et le relâchement d'un "*droit de parler*" également appelé "*jeton*". Il n'y a pas de phase où le téléphone sonne pour prévenir l'arrivée d'un message, les utilisateurs sont toujours supposés être prêts à répondre durant une session.

Le Push To Talk over Cellular (PoC) fonctionne à travers les réseaux GSM, GPRS, UMTS

FIG. 2.1 – Le Push To Talk.

Une conversation Push To Talk consiste à presser un bouton et à parler. Le message vocal est diffusé à tous les membres de la conférence. Il existe différentes façons de créer une conférence :

Le mode LOGIN Dans ce mode, la conférence est supposée connue par ses membres, celle-ci est identifiée par un nom (qui est une chaîne de caractères). A n'importe quel

moment de la conférence un membre peut se connecter à celle-ci. La conférence peut être assimilée à un "*salon public*" où tout le monde peut venir se connecter.

Le mode INVITE En mode INVITE, la conférence est créée par son premier membre. Celui-ci la crée avec une liste de membres. Une invitation (au sens SIP du terme) est envoyée à chaque membre de la liste. Lorsqu'ils ont accepté ou refusé l'invitation, la conférence est créée.

Le mode MULTIGROUP Lors d'un échange en mode MULTIGROUP, la conférence est initiée lors du premier message d'un utilisateur vers un ou plusieurs utilisateurs. Il s'agit ici d'une interaction élémentaire. La création de la conférence est transparente du point de vue de l'utilisateur.

Le service Push To Talk utilise le protocole SIP pour la signalisation des messages (pour la création de conférences par exemple). L'utilisateur dispose d'une liste de contacts qui lui permet de sélectionner les différents utilisateurs avec qui il désire communiquer. Cette liste de contacts indique en temps réel l'information de présence d'un utilisateur qui est obtenue par un serveur de présence. Nous expliquerons plus en profondeur le serveur de présence dans la troisième partie de ce document.

Les messages vocaux sont, eux, transportés par le protocole RTP une fois la signalisation effectuée au niveau SIP. Il y a séparation entre la signalisation et le transport des données comme c'est le cas dans les systèmes Voice Over IP.

Après cette brève présentation, nous allons maintenant regarder plus en profondeur le fonctionnement des différents modes de conversation. Mais avant cela, toutes les interactions Push To Talk étant basées sur des requêtes SIP, nous devons traiter la phase préliminaire nécessaire à des échanges de messages, à savoir l'enregistrement.

2.1 L'enregistrement

Lorsqu' un utilisateur ouvre son application cliente Push To Talk, il s'enregistre auprès du système. Cet enregistrement correspond à l'envoi d'une requête SIP *REGISTER*. Elle a pour but d'associer l'adresse IP du terminal de l'utilisateur à sa SIP URI.

Notons qu'à ce stade s'effectue également un processus d'authentification qui permet d'autoriser ou non l'utilisateur à s'enregistrer au système.

La figure 2.2 nous montre le diagramme de séquence relatif à l'enregistrement

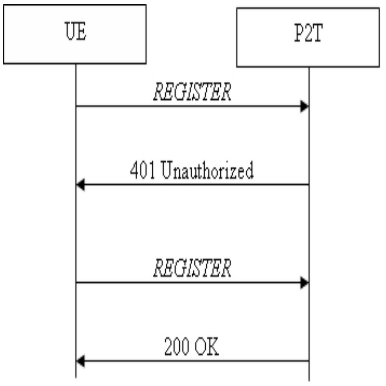


FIG. 2.2 – Enregistrement au système Push To Talk.

2.2 Initiation d'une conférence

2.2.1 Initiation du mode LOGIN

En mode LOGIN, une requête SIP *INVITE* est envoyée au serveur Push To Talk par le créateur du "*salon public*". Ce message est suffisant pour créer la conférence. Une fois cette conférence créée, n'importe quel utilisateur peut venir s'y connecter en envoyant également un message SIP *INVITE* avec le nom du "*salon public*".

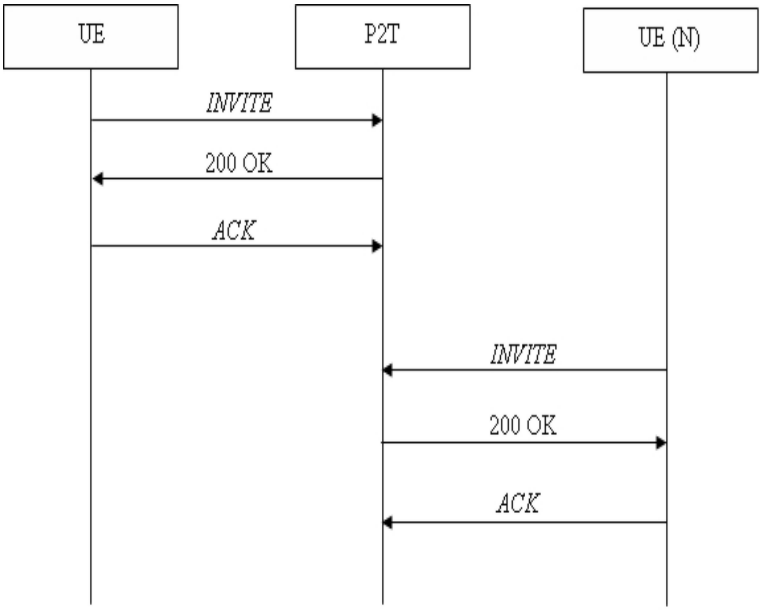


FIG. 2.3 – Initiation d'une conférence en mode LOGIN.

2.2.2 Initiation du mode INVITE

En mode INVITE, l'initiateur de la conférence envoie une requête SIP *INVITE* au serveur Push To Talk. L'initiateur a sélectionné au préalable une liste de récipients. Pour

transporter cette liste deux possibilités sont offertes. Soit :

1. Un header est fournit dans le message SIP indiquant où trouver la liste.
2. Le corps du message SIP est de type "*multipart*" et contient la liste des récipients et le document SDP nécessaire à la description de la session.

La deuxième solution a été préférée par l'organisme de standardisation OMA.

Lorsque le serveur Push To Talk reçoit le message *INVITE*, il récupère la liste de récipients et envoie à son tour une requête *INVITE* secondaire à tous les récipients appartenant à cette liste. Ces récipients ont alors le choix entre accepter ou refuser l'invitation. Lorsqu'ils ont répondu, la conférence est créée, les membres de cette conférence peuvent avoir la parole.

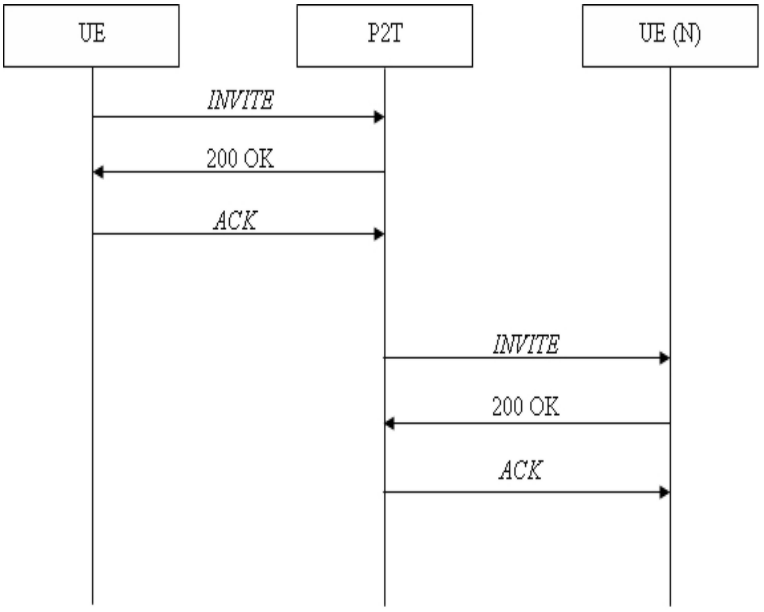


FIG. 2.4 – Initiation d'une conférence en mode INVITE.

2.2.3 Initiation du mode MULTIGROUP

En mode MULTIGROUP, chaque utilisateur se connecte individuellement au serveur Push To Talk.

Dans ce mode, la requête *INVITE* est typiquement envoyée au démarrage de l'application cliente (au moment de l'enregistrement). Elle ne contient aucun attribut supplémentaire tel qu'un nom de "*salon public*" dans le cas du mode LOGIN ou une liste de membres dans le cas du mode INVITE.

Nous venons de montrer les différentes façons d'initier des conférences. Le but de ces conférences est de permettre aux utilisateurs d'interagir entre eux. Comme nous l'avons vu

plus haut, cette interaction se fait par la réservation d'un "*jeton*". Nous allons maintenant montrer comment est réalisée la réservation du "*jeton*".

2.3 Réservation du "*jeton*"

La réservation ou le relâchement du "*jeton*" s'effectue à l'aide d'un message SIP *INFO* dont le header "*subject*" contient une commande qui est soit :

- reserve : essayer de réserver le "*jeton*" ;
- reserved : annoncer que le "*jeton*" est réservé ;
- free : libérer un "*jeton*" réservé.

Si l'utilisateur désire réserver le "*jeton*" alors qu'il l'est déjà, il reçoit comme réponse un code d'erreur 403 (FORBIDDEN). Lorsque le "*jeton*" est libre, l'utilisateur reçoit un code 200.

Lorsqu'un terminal s'est vu attribuer le "*jeton*", les terminaux appartenant à la même conférence sont prévenus de la réservation par un message SIP *INFO* dont le header "*subject*" contient le mot *reserved*. Ils s'attendent alors à recevoir un message vocal.

Lorsque l'utilisateur ayant réservé le "*jeton*" reçoit un code 200 en réponse à sa requête SIP *INFO*, un message RTCP *SOS* (Start Of Speech) est envoyé à tous les récipients de la conférence.

Le même principe est appliqué lors de la relâche du "*jeton*". Un message SIP INFO dont le header "*subject*" contient le mot *free* est envoyé à tous les récipients. Un message RTCP *EOS* (End Of Speech) est également envoyé.

La figure 2.5 nous montre le diagramme de séquence relatif à la réservation du "*jeton*".

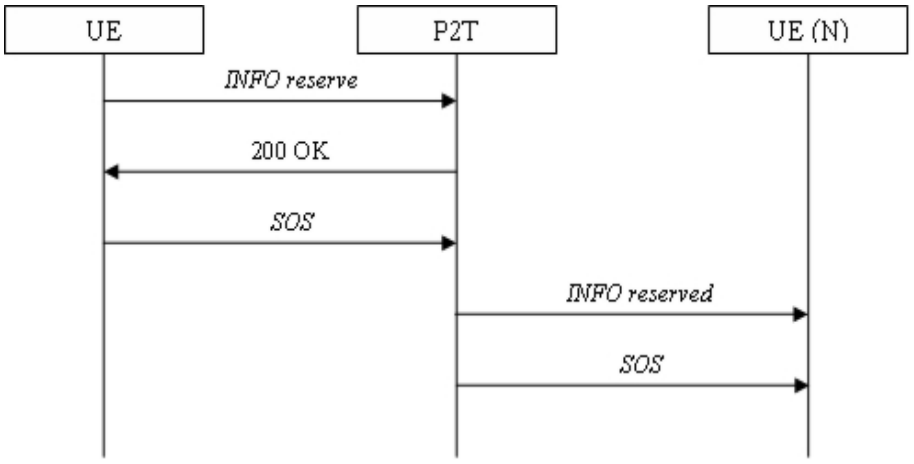


FIG. 2.5 – Réservation du jeton.

L'utilisateur désirant réserver le "*jeton*" envoie une requête SIP *INFO* dont le header "*subject*" contient le mot *reserve*. Le serveur Push To Talk confirme la réservation

du "*jeton*" par une réponse ayant pour code 200. Ensuite il prévient les récipients de la réservation du "*jeton*" par un message SIP *INFO* dont le header contient le mot *reserved* et un message RTCP *SOS*.

La figure 2.6 nous montre le diagramme de séquence relatif au relâchement du "*jeton*".

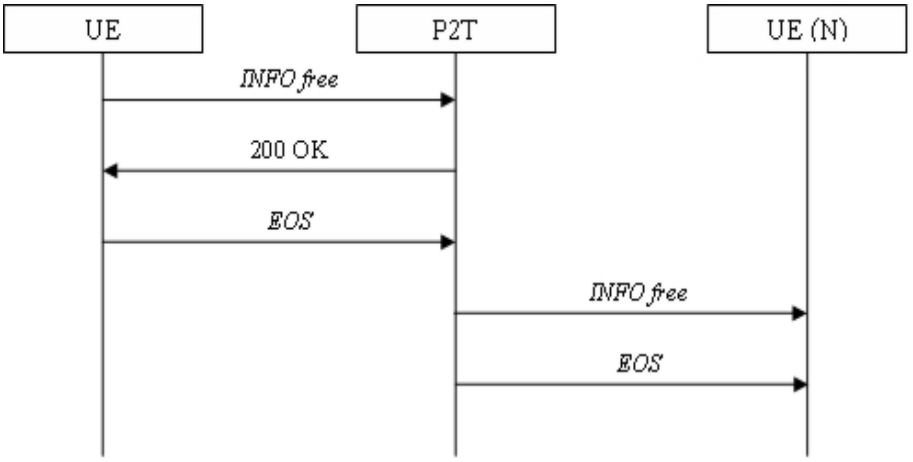


FIG. 2.6 – Relâchement du jeton.

L'utilisateur désirant relâcher le "*jeton*" envoie une requête SIP *INFO* dont le header "*subject*" contient le mot *free*. Le serveur Push To Talk confirme la réservation du "*jeton*" par une réponse ayant pour code 200. Ensuite il prévient les récipients de la disponibilité du "*jeton*" par un message SIP *INFO* dont le header contient le mot *free* et un message RTCP *SOS*.

Après s'être introduit dans une conférence et avoir participé à celle-ci, il est logique de pouvoir s'en retirer ou de mettre fin à la conférence. C'est l'objet de notre prochaine section.

2.4 Fin d'une session

Pour se retirer d'une session l'utilisateur envoie une requête SIP *BYE*.

La figure 2.7 nous montre le diagramme de séquence relatif à la fin d'une session. Il est valable pour tous les modes de conversation mais a des incidences différentes suivant la personne émettrice du message ou le type de conversation.

Dans le cas d'une conférence en mode LOGIN, l'envoi d'une requête SIP *BYE* par utilisateur ne clôt la conférence que si cet utilisateur est le dernier présent dans le "*salon public*".

Dans le cas d'une conférence en mode INVITE, l'envoi d'une requête SIP *BYE* par l'initiateur de la conférence a pour conséquence de terminer la conférence. Le serveur Push To Talk envoie à son tour une requête SIP *BYE* secondaire aux autres utilisateurs participant

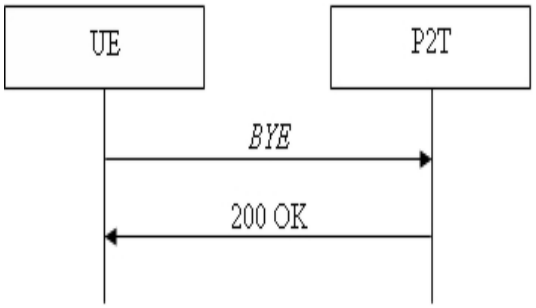


FIG. 2.7 – Fin de session.

à la conférence pour les prévenir de la clôture de celle-ci.

En mode MULTIGROUP, l'envoi d'un message SIP *BYE* signifie que l'utilisateur n'est plus disposé à recevoir aucun message Push To Talk, ce message est donc envoyé typiquement lorsque l'utilisateur éteint son application.

Chapitre 3

La *Proxy Platform*

Nous avons effectué notre stage au sein de la société *Nextenso.S.A*. La plate-forme de développement sur laquelle se basent les applications de *Nextenso.S.A* s'appelle la *Proxy Platform*. La présentation de cette plate-forme s'insprire des références [11, 10]. La *Proxy Platform* permet de développer des solutions logicielles orientées réseau. Elle se place dans un flux réseau entre un client et un serveur et agit à la manière d'un proxy.

Avec la *Proxy Platform*, deux types de tâche peuvent essentiellement être réalisés :

1. Contrôler et modifier le flux échangé entre un client et un serveur. Ceci est réalisé grâce aux proxylets (que nous expliquerons plus en détail au point 3.2)
2. Résoudre des problèmes de compatibilité qui interviennent lorsqu'un client et un serveur n'utilisent pas le même protocole.

Exemple :

Les téléphones mobiles n'utilisent pas le protocole HTTP mais le protocole WAP. Dès lors si un téléphone mobile veut se connecter à un serveur HTTP, il ne pourra le faire que par l'intermédiaire d'une WAP Gateway, c'est à dire une passerelle entre le protocole WAP et le protocole HTTP.

3.1 Les caractéristiques de la *Proxy Platform*

Les principales caractéristiques de la *Proxy Platform* sont les suivantes :

- la *Proxy Platform* est *scalable*[1] et permet d'écrire des solutions logicielles possédant cette caractéristique. Toute application basée sur la *Proxy Platform* peut être instanciée sur une ou plusieurs machines en augmentant par la même occasion sa capacité de traitement ;
- la *Proxy Platform* permet de faire de la répartition de charge[2], c'est à dire que lors-

[1] Le lecteur voudra bien nous excuser de n'avoir pu trouver de traduction française satisfaisante.
[2] *load balancing*.

qu'une application est instanciée sur plusieurs machines, les agents qui la constituent voient la charge répartie équitablement ;
- La *Proxy Platform* est administrable via une interface web. Il est possible de commander et configurer les applications à partir d'un client HTTP. Ainsi, lorsqu'une application est instanciée sur plusieurs machines, les instances peuvent être activées ou désactivées grâce à l'interface web.

3.2 Les composants de la *Proxy Platform*

La *Proxy Platform* abrite 3 composants :

La stack C'est elle qui intercepte le flux réseau de bas niveau entre le client et le serveur. Elle est écrite en C pour pouvoir gérer les connexions de manière performante. Une stack est propre à un protocole (stack HTTP, stack WAP,...).

Le callout agent Il reçoit le flux réseau intercepté par la stack. Les données reçues sont transformées à l'aide de proxylets, le callout agent est donc un conteneur de proxylets. Une fois les données reçues transformées, l'agent peut soit les rediriger vers la stack d'origine, soit les envoyer vers une autre stack de protocole différent. Ce cas est utilisé pour passer d'un protocole à un autre.

Exemple : Dans le cas d'une WAP Gateway, le callout agent reçoit le trafic d'une WAP stack et le redirige vers un HTTP stack.

La proxylet Il s'agit d'un morceau de code java permettant de traiter le flux réseau de haut niveau. Les proxylets peuvent être chaînées (placées l'une à la suite de l'autre sur un flux), ce qui permet de découper un problème en plusieurs sous-problèmes plus simples. Ce chaînage se fait sur deux flux : le flux de demande et le flux de réponse. Il est donc possible d'écrire des applications extrêmement modulables. Notons que le chaînage des proxylets est dynamique : à la fin de son exécution, une proxylet a le choix entre transmettre le flux à la proxylet suivante, à la première proxylet du flux de réponse (si cette proxylet est dans le flux de requête, dans ce cas elle court-circuite la chaîne de requête) ou à court-circuiter toute la chaîne (c'est à dire envoyer le flux après la dernière proxylet de réponse). Une proxylet peut également choisir de traiter ou non le flux qu'elle reçoit.

Il existe deux types de proxylets : les *streamed proxylets* et les *buffered proxylets*. Contrairement aux *streamed proxylets*, les *buffered proxylets* retiennent le flux pour qu'il soit traité. Celles-ci sont moins performantes, mais il n'est pas toujours possible de traiter un flux en continu.

Notons que pour optimiser ses performances, la *Proxy Platform* utilise son propre protocole de communication entre un callout agent et une stack, le MUX. Ce protocole permet de réaliser le mutiplexage de plusieurs connexions à travers une seule socket TCP.

3.3 Exemple d'utilisation de la *Proxy Platform*

Afin de mieux comprendre les notions explicitées au début de ce chapitre, nous allons utiliser un exemple qui est illustré à la figure 3.1.

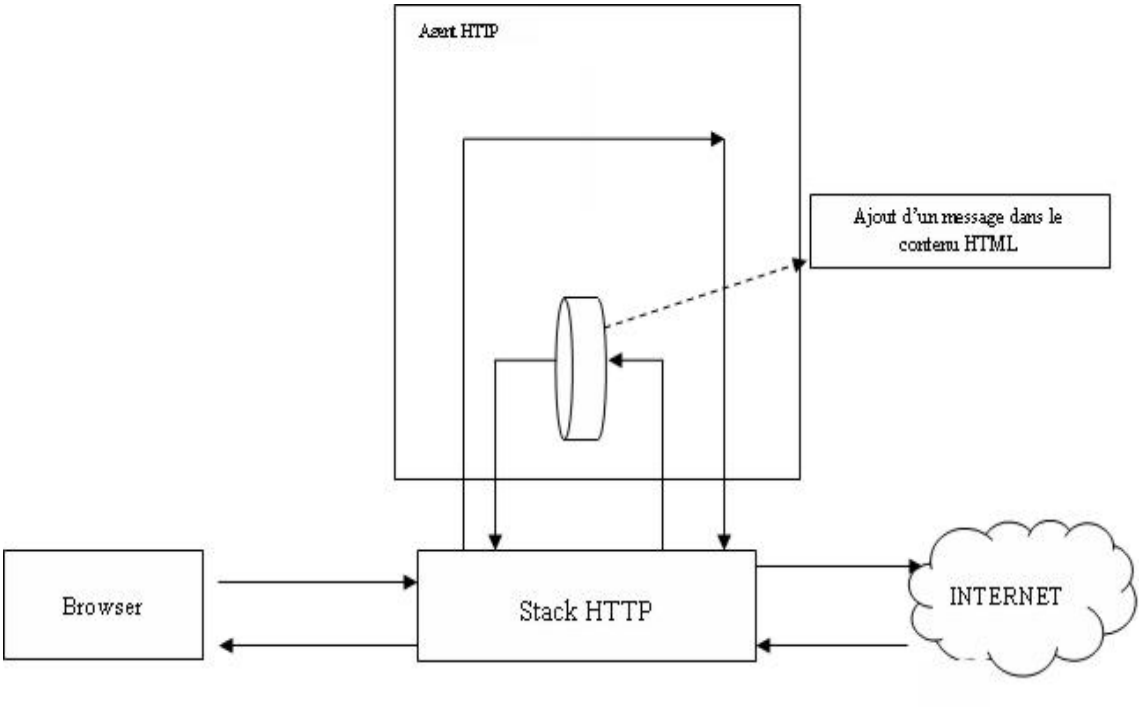


FIG. 3.1 – Exemple d'utilisation de la *Proxy Platform*.

Supposons que nous soyons fournisseur d'accès Internet (FAI). Nous aimerions diffuser un message à tous nos clients. Nous aimerions que nos clients lisent le message dès qu'ils se connectent à Internet, quel que soit le site qu'ils aient choisi.

Pour ce faire, nous avons besoin d'une stack HTTP, au-dessus de laquelle nous placerons un agent HTTP qui contiendra une proxylet, que nous placerons sur le flux de réponse. Cette proxylet aura pour but d'ajouter le message au début de la page HTML contenue dans le corps de la réponse HTTP.

Pour illustrer la modularité des proxylets, supposons maintenant que nous voulions offrir un service de traduction de pages à nos clients. Il nous suffit d'ajouter une proxylet dans le flux de réponse qui a pour but de traduire la page HTML demandée par le client.

Après cette brève (et nécessaire) introduction à la *Proxy Platform*, nous allons maintenant pouvoir entrer dans le détail des architectures.

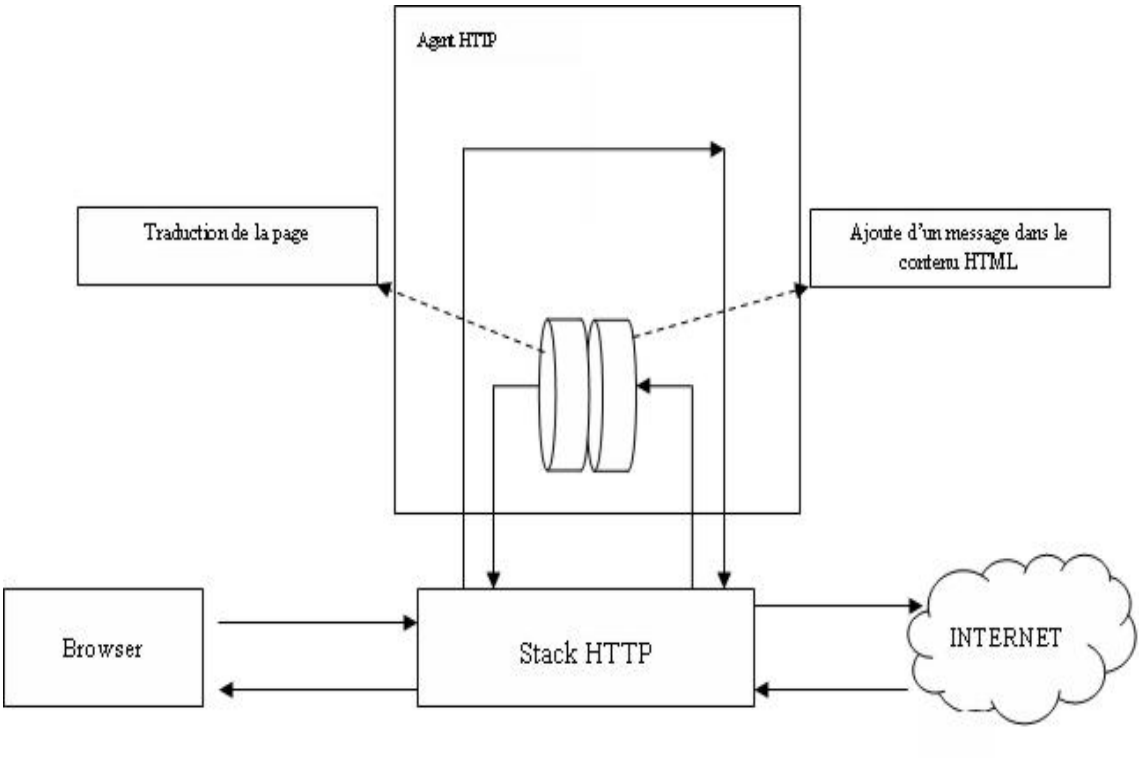


FIG. 3.2 – Illustration de la modularité de la *Proxy Platform*.

Deuxième partie

SMS/MMS Gateway

Chapitre 4

Les services SMS et MMS Premium

4.1 But et environnement

C'est à l'occasion d'une élection de *Miss Belgique* que le SMS Premium a vu le jour en 2000 en Belgique. Les téléspectateurs pouvaient voter pour leur candidate préférée en envoyant un SMS. Le succès de ce service a permis aux opérateurs de se rendre compte du potentiel énorme qu'il représentait.

Le SMS Premium (ou SMS surtaxés, ces messages pouvant être facturés entre 0,15 et 4 euros) est un service offert aux utilisateurs de téléphones mobiles, leur permettant d'obtenir de l'information par l'envoi au préalable d'un SMS à un numéro court (en Belgique, il s'agit de numéros à 4 chiffres commençant par un 3) contenant un ou plusieurs mots-clés.

Ces services rencontrent un franc succès et représentent pour l'opérateur belge *Mobistar* 4% des 2,5 millions de SMS envoyés chaque jour sur le réseau. Ils sont par ailleurs générateurs d'un chiffre d'affaires de plusieurs dizaines de millions d'euros par an[1]. Pour s'en convaincre, il suffit de regarder autour de soi. Nombreuses sont les publicités, que ce soit à la télévision, dans les magazines ou même sur Internet, qui font la promotion de services tels que : recevoir l'actualité, la météo, les résultats sportifs, ...

Le MMS Premium est au MMS ce que le SMS Premium est au SMS. Compte tenu des qualités du MMS, notamment la richesse du contenu qui peut être multimédia (texte, image, son, vidéo), on imagine facilement l'intérêt que lui prêtent les opérateurs pour ce genre de services.

La figure 4.1 nous montre les 3 acteurs du marché du SMS MMS Premium, ainsi que leur relation :

L'opérateur C'est lui qui héberge la SMS/MMS Gateway, il se contente de la faire fonctionner dans son parc informatique sans se soucier du contenu apporté par les dif-

[1] Selon [13].

férents services. L'opérateur sert d'intermédiaire entre le fournisseur de contenu et l'utilisateur final. Ses revenus proviennent essentiellement du trafic de SMS et de MMS généré par les différents services. Le fournisseur de contenu devra également rémunérer l'opérateur lors de la création d'un service.

Le fournisseur de contenu Il fournit le contenu associé aux différents services dans lesquels il est impliqué. Celui-ci sera disponible sous la forme de SMS ou de MMS. L'avantage du MMS est qu'il n'est pas limité en nombre de caractères comme un SMS. Il permet également d'insérer du contenu multimédia plus riche et donc générateur de revenus plus élevés. Pour chaque service, le fournisseur de contenu a la choix entre :

- laisser son contenu chez l'opérateur ;
- le rendre disponible sur un serveur HTTP.

Les revenus du fournisseur de contenu proviennent d'un pourcentage des revenus perçus par l'opérateur sur le trafic généré par les différents services.

L'utilisateur final Pour bénéficier des services, il doit envoyer un SMS contenant un mot-clé à un numéro spécifique. Exemple : l'utilisateur envoie TDF au 3310 et il reçoit le classement de l'étape du jour du Tour de France.

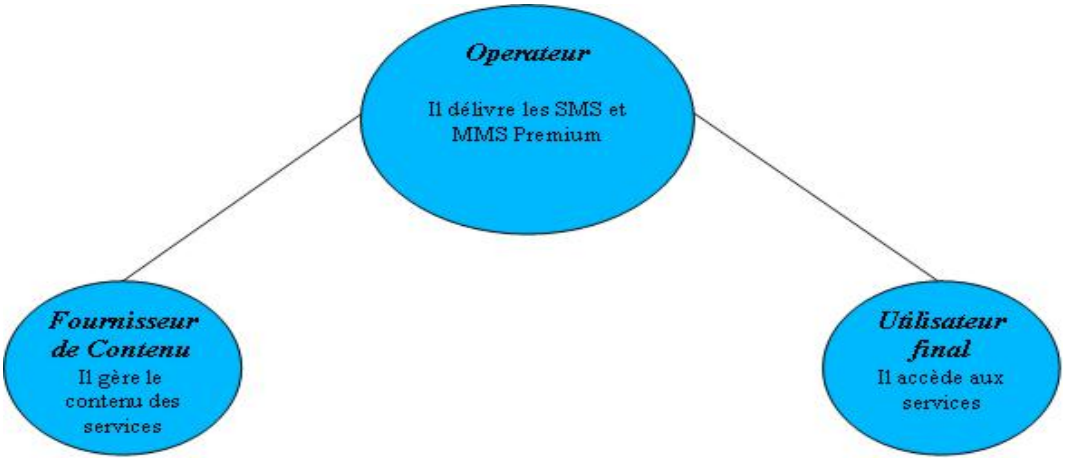


FIG. 4.1 – Acteurs du marché du SMS MMS Premium.

4.2 La SMS Gateway

La SMS Gateway est un produit développé par la société *Nextenso S.A.* Il s'agit d'une solution complète pour un opérateur de communications mobiles qui lui permet de gérer des services SMS Premium en partenariat avec un ou plusieurs fournisseurs de contenu.

En terme de *Proxy Platform*, il s'agit d'un agent SMS contenant une proxylet qui a principalement pour but de récupérer le(s) mot(s) clé(s) se trouvant dans le message, trouver la correspondance au(x) mot(s) clé(s) et fournir la réponse adéquate à l'utilisateur final.

La figure 4.2 illustre l'environnement de la SMS Gateway.

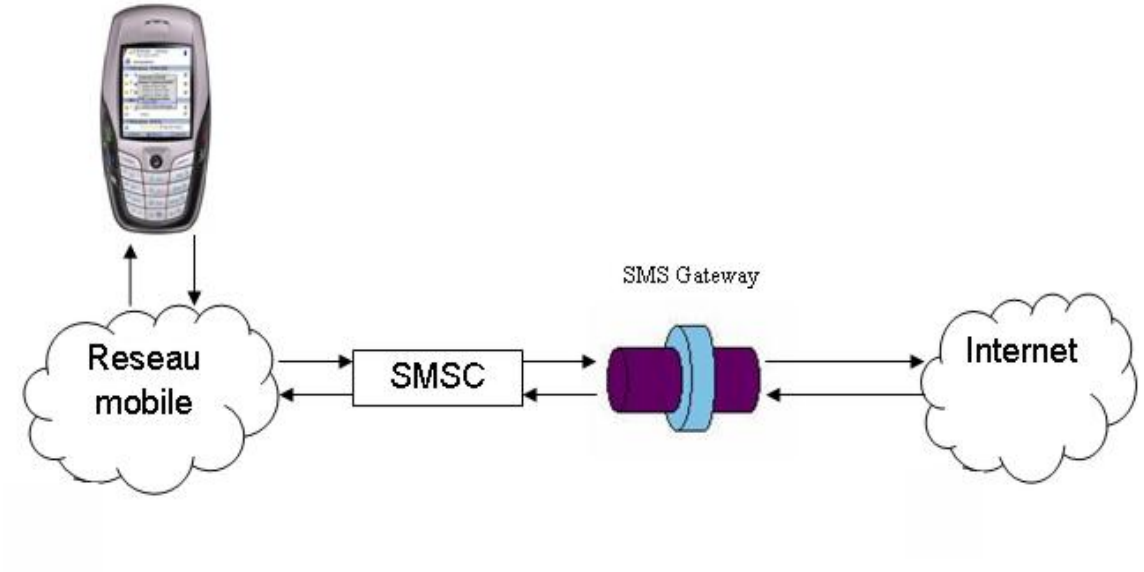


FIG. 4.2 – Environnement de la SMS Gateway.

Le processus de l'envoi d'un SMS Premium par l'utilisateur à la réception de la réponse est composé de 4 étapes :

1. **Envoi de la requête** : l'utilisateur envoie un SMS contenant un ou plusieurs mots-clés
2. **Recherche du mot-clé** : la SMS Gateway récupère le contenu du message et vérifie s'il correspond à un des *mappings* de sa base de données.
3. **Récupération du contenu de la réponse** : une fois le *mapping* identifié, la SMS Gateway doit récupérer le contenu correspondant. Pour ce faire, soit elle envoie une requête à la destination correspondante (applications, site web, fichier interne), soit elle récupère un texte prédéfini dans sa base de données.
4. **Envoi de la réponse** : la SMS Gateway reçoit la réponse de sa requête ou récupère un texte prédéfini et envoie le contenu sous forme de SMS à l'utilisateur final.

Analysons de plus près ces 4 étapes.

4.2.1 Envoi de la requête

Les paramètres de la requête dont nous avons besoin sont les suivants :
- le numéro de téléphone de l'expéditeur ;
- le numéro de téléphone du destinataire ;
- un ou plusieurs mots-clés, par exemple : "*météo paris*". Ce mot-clé correspond à un *mapping* qui permet de retrouver le contenu de la réponse à fournir.

4.2.2 Recherche du mot-clé

Il faut tout d'abord identifier le bon *brand*. La notion de *brand*, ou plus exactement de "*multi-branding*", peut être définie comme la capacité de déployer plusieurs portails sur un même serveur. Dans notre cas, les *brands* sont utilisés pour permettre à des utilisateurs appartenant à différents opérateurs d'accéder au même serveur. Un *brand* va donc permettre de filtrer un SMS suivant des critères prédéfinis. Prenons par exemple comme critère le numéro de téléphone de l'expéditeur ou du destinataire (*brand* basé sur une adresse). Nous pouvons dans ce cas définir le brand #1 auquel nous associons tous les numéros d'appel commençant par +32047. De la même façon nous pouvons définir le *brand* #2 auquel nous associons tous les numéros de destination égaux à 3310.

Il faut ensuite retrouver le *mapping* à appliquer. Un *mapping* est associé à un ou plusieurs *brand* (de la même façon un *brand* est associé à un ou plusieurs *mappings*). La SMS Gateway va parcourir tous les *mappings* relatifs au *brand* identifié grâce au(x) mot(s)-clé(s) contenu(s) dans le message. Lorsque le *mapping* est identifié le contenu de la réponse à envoyer à l'utilisateur final est récupéré. Il existe trois types de *mapping* :

1. **Les *mappings "mot-clé"*** : avec ces *mappings* simples, il est possible d'associer une URL prédéfinie. Par exemple, si au mot-clé "*météo*" est associé une URL `file://meteo.txt`. Lorsque la SMS Gateway recevra comme message le mot-clé "*météo*", celle-ci récupérera le texte contenu dans le fichier `meteo.txt`. Ce type de *mapping* est assez limité. Ainsi, lorsque nous souhaitons fournir la météo dans différentes villes, nous devons définir autant de *mappings* que de villes pour lesquels nous souhaitons fournir la météo. C'est pourquoi il existe un autre type de *mappings* plus complexe ...
2. **Les *mappings "expression régulière"*** : grâce aux expressions régulières, il est possible d'associer plusieurs mots-clés à un *mapping*. Par exemple l'expression régulière suivante permettra de définir un *mapping* qui aura pour but de fournir la météo dans différentes villes :

   météo \s+(\S+)$

   \s correspond à un espace

   + correspond à une ou plusieurs fois le caractère donné (espace)

   \S correspond à un caractère différent d'un espace

   (\S+) correspond à un groupe composé d'un ou plusieurs caractères sauf un espace

   $ correspond à la fin du contenu du SMS

   A cette expression régulière pourra être associée une URL `file://$1.txt` où $1 sera remplacé par le contenu de (\S+).
   Lorsque l'utilisateur enverra un SMS contenant "*météo paris*", la SMS Gateway récupérera le contenu à l'URL `file://paris.txt`. De cette façon, si nous souhaitons

fournir la météo pour une nouvelle ville, il ne sera pas nécessaire de créer un nouveau *mapping* mais simplement d'ajouter le fichier correspondant à cette ville.

3. **Les *mappings* "*compteur*"** : ces *mappings* sont typiquement utilisés lorsqu'un utilisateur désire voter pour un candidat à la télévision. Aucune réponse n'est envoyée à l'utilisateur final.

4.2.3 Récupération du contenu de la réponse

La récupération du contenu se fait grâce à une URL. Celle-ci peut être :
- une requête HTTP : `http://www.yahoo.com/weather?city=paris`, la SMS Gateway renverra comme réponse le contenu du corps de la réponse HTTP ;
- une requête `fichier://meteo.txt`, la SMS Gateway renverra comme réponse le contenu du fichier meteo.txt ;
- une requête texte : `text://mon_texte`, il s'agit d'un texte défini statiquement et qui sera renvoyé par la SMS Gateway. Si nous associons à un *mapping* l'URL `text://il_fera_beau_demain`, l'utilisateur final ayant souscrit au service recevra en réponse un SMS contenant le texte "*il fera beau demain*".

4.2.4 Envoi de la réponse

La réponse obtenue à l'URL correspondant au *mapping* constitue la réponse du SMS. Si cette réponse est plus importante que la taille maximale d'un SMS, elle sera séparée en plusieurs SMS, le nombre maximum de SMS par réponse étant configurable.

Chapitre 5

La SMS/MMS Gateway

La SMS/MMS Gateway est une évolution de la SMS Gateway, son but étant de ne plus limiter les services au SMS Premium mais d'offrir la possibilité de gérer les MMS Premium.

Pour des raisons commerciales, la société *Nextenso S.A* souhaitait ne pas modifier la SMS Gateway, celle-ci étant un produit fini et vendu. Cette volonté nous contraignait à ajouter un élément dans l'environnement qui soit transparent aux yeux de la SMS Gateway. De cette façon, lorsqu'un client possède la passerelle, le système ne doit pas être réinstallé. Il suffit de greffer un élément supplémentaire, ce qui lui permet de garder toute sa configuration intacte.

Nous avons profité du fait que la SMS Gateway permet d'envoyer des requêtes HTTP en plaçant un agent HTTP comme proxy de celle-ci. Grâce à un format précis donné aux requêtes correspondant à un service MMS Premium, l'agent HTTP peut distinguer les requêtes HTTP qui lui sont destinées des autres.

Le rôle de l'agent HTTP est :
- d'intercepter les requêtes émanant de la SMS Gateway si celles-ci correspondent à un service MMS Premium ;
- construire le MMS ;
- envoyer le MMS au format MM7.

Pourquoi le format MM7 ?

Comme nous l'avons expliqué dans l'introduction, l'interface MM7 permet à une application (appelée VAS) d'interagir indirectement avec un utilisateur MMS en passant par le MMSC. Dans ce genre de scénario appelé "*machine à personne*" le VAS envoie une requête *MM7_submit.REQ* à destination du MMSC. Celui-ci va envoyer le message à tous les destinataires par l'interface MM1.

La SMS/MMS Gateway joue dans ce cas le rôle du VAS, comme décrit à la figure 5.1 elle envoie une requête *MM7_submit.REQ* au MMSC afin que celui-ci l'envoie à un ou plusieurs destinataires au format MM1. Les utilisateurs seront notifiés de la réponse par la

requête *MM1_ notification.REQ*.

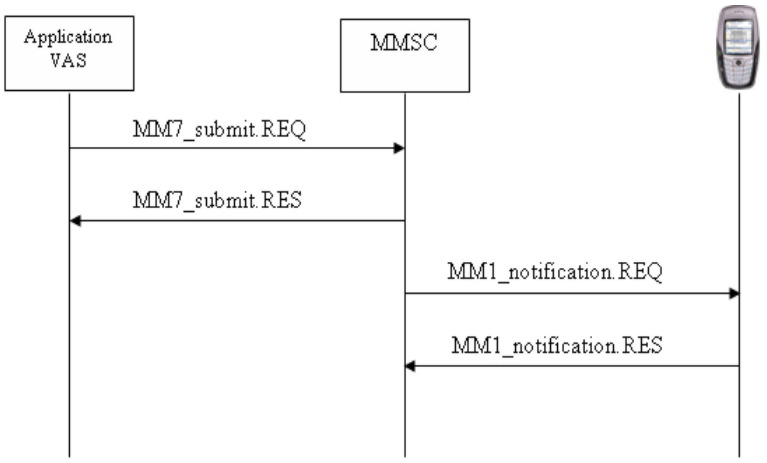


FIG. 5.1 – Envoi de la réponse à l'utilisateur final (inspiré de [3, page 242]).

Nous avons ajouté la notion de "compte de fournisseur". Lorsqu'un fournisseur de contenu désire fournir un service MMS Premium, il doit ouvrir un compte chez l'opérateur.

La SMS/MMS Gateway met à la disposition des fournisseurs de contenu deux types de service MMS Premium :

1. **Un service *MM1 In Response*** où le fournisseur de contenu s'engage à fournir un MMS (au format MM1) disponible à une adresse HTTP. Notre agent HTTP devra simplement télécharger le MMS et l'adapter au format MM7 pour l'envoyer à l'utilisateur final.
2. **Un service *MMSFactory*** où le MMS à envoyer en réponse est hébergé chez l'opérateur sous forme d'un *template*.

Une interface graphique[1] a été implémentée. Elle permet à l'opérateur de gérer les "comptes de fournisseur". Lorsqu'un fournisseur de contenu ouvre un compte chez l'opérateur, celui-ci lui fourni un nom d'utilisateur et un mot de passe. Grâce à cela, le fournisseur de contenu peut se connecter à son compte et gérer ses différents services MMS.

Cette interface est particulièrement utile pour les services *MMSFactory* puisqu'elle offre la possibilité au fournisseur de contenu de construire le MMS qu'il désire fournir en réponse à l'utilisateur final tout en lui offrant un aperçu de celui-ci. L'interface lui permet de définir le contenu de chaque page. Un page est composée de deux éléments. L'insertion d'un texte peut se faire de deux façons :

1. Le fournisseur de contenu peut entrer le texte de manière statique.
2. Il peut entrer l'URL d'un fichier XML accompagné d'un XPATH permettant de retrouver le noeud du fichier XML contenant le texte.

[1] Interface web implémentée en JSP et hébergée chez l'opérateur.

L'insertion d'un élément multimédia peut également se faire de deux façons :

1. Le fournisseur de contenu peut entrer une URL indiquant où récupérer l'élément multimédia.
2. Il peut entrer l'URL d'un fichier XML accompagné d'un XPATH. Cet XPATH permettra de retrouver le noeud du fichier XML contenant l'URL indiquant où télécharger l'élément multimédia.

L'avantage du fichier XML est qu'il permet au fournisseur de contenu de modifier son MMS sans se connecter à la machine de l'opérateur et occuper les ressources de celle-ci.

Analysons de plus près les composants de la SMS/MMS Gateway.

5.1 Architecture de la SMS/MMS Gateway

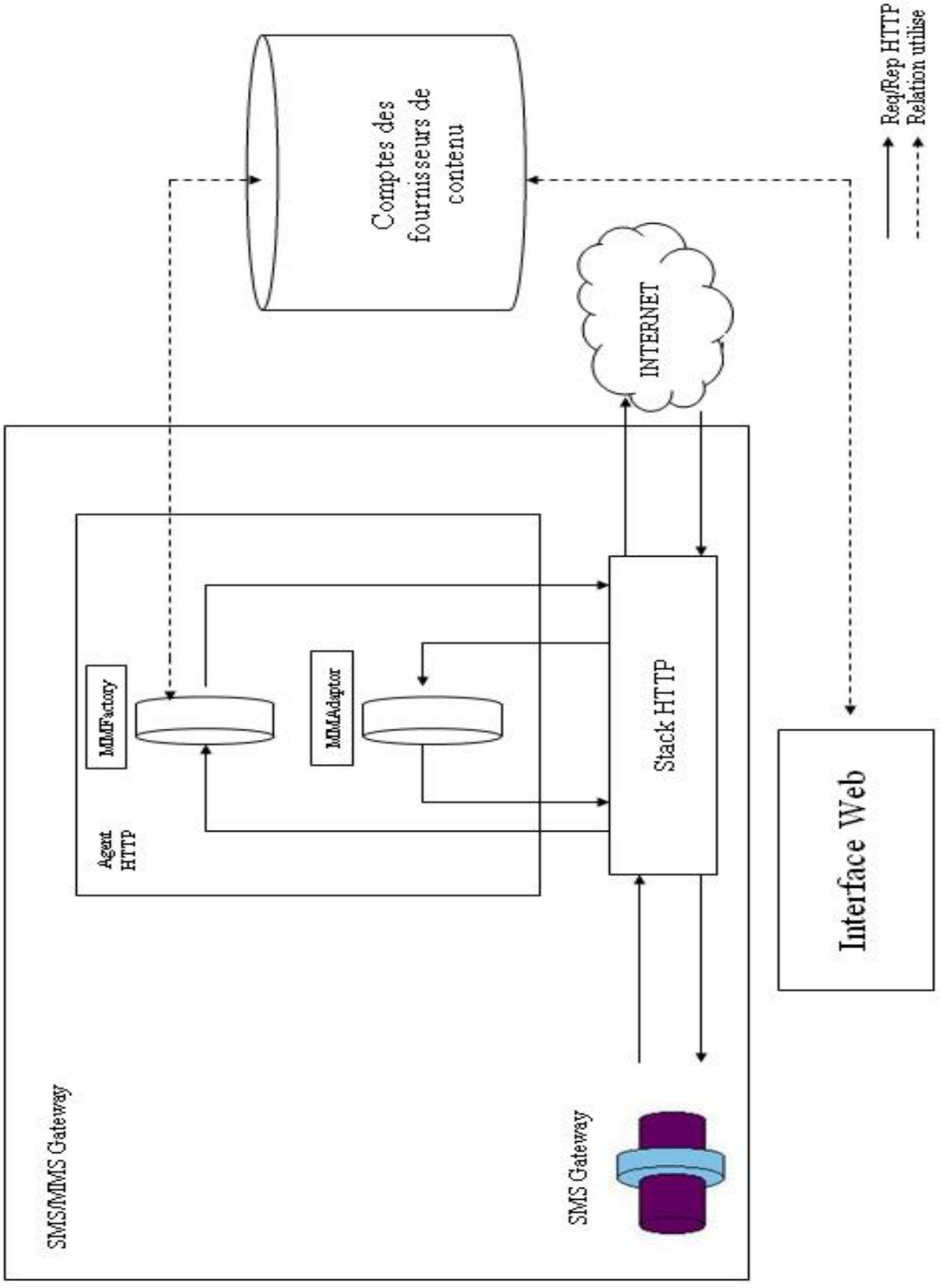


FIG. 5.2 – Architecture de la SMS/MMS Gateway.

Comme nous l'avons déjà précisé, la gestion des MMS Premium se fait par l'intermédiaire d'un agent HTTP. Cet agent se place derrière la SMS Gateway et intercepte ses requêtes HTTP. Il est constitué de deux proxylets, une sur le flux de requête (*MMFactory*), l'autre sur le flux de réponse (*MMAdaptor*).

Le but de la proxylet *MMFactory* est de récupérer le compte du fournisseur de contenu et le service concerné. Elle vérifie ensuite le type de services :

- s'il s'agit d'un service de type ***MM1 In Response***, elle envoie une requête HTTP pour récupérer le MMS au format MM1 ;
- s'il s'agit d'un service de type ***MMSFactory***, elle récupère le *template* du MMS et construit celui-ci au format MM1[2].

Enfin, elle court-circuite la chaîne de requête pour passer la main à la proxylet *MMAdaptor*.

La proxylet *MMAdaptor* reçoit le message au format MM1, l'encode au format MM7 et l'envoie à l'utilisateur final.

L'agent HTTP ne peut traiter les flux qui concernent les services SMS Premium. Dès lors, la proxylet *MMFactory* doit distinguer les requêtes qui concernent les services MMS Premium pour les traiter et laisser passer les autres flux sans les traiter.

Pour rappel, une proxylet peut choisir de traiter ou pas un flux. Pour cela il faut trouver un critère qui déclenche le traitement de celui-ci. Nous avons choisi comme critère le format de l'URL de la requête HTTP. La proxylet ne traitera le flux intercepté que si l'URL a le format suivant :

```
http://mms?provider=providerName&service=serviceName
```

Le même problème est rencontré lors de la réception de la réponse HTTP. La proxylet *MMAdaptor* ne doit traiter que les réponses qui concernent les services MMS Premium et laisser passer les autres. Pour remédier à ce problème nous avons choisi de placer un header "*X-mms_response*" dans la *MMFactory* lorsque celle-ci traite la requête. La proxylet *MMAdaptor* ne traitera que les réponses possédant un tel header.

L'illustration de l'architecture que nous venons d'expliquer se trouve à la figure 5.2.

Nous allons maintenant regarder de plus près l'interaction entre les différents composants à l'aide de diagrammes de séquence.

5.2 Interaction entre les différents composants

Nous allons dans cette section analyser tous les scénarii possibles à l'aide de diagrammes de séquence que nous commenterons.

[2]C'est ici que se situe la partie la plus complexe et la plus intéressante d'un point de vue algorithmique.

5.2.1 Envoi d'un SMS Premium et réception d'un SMS

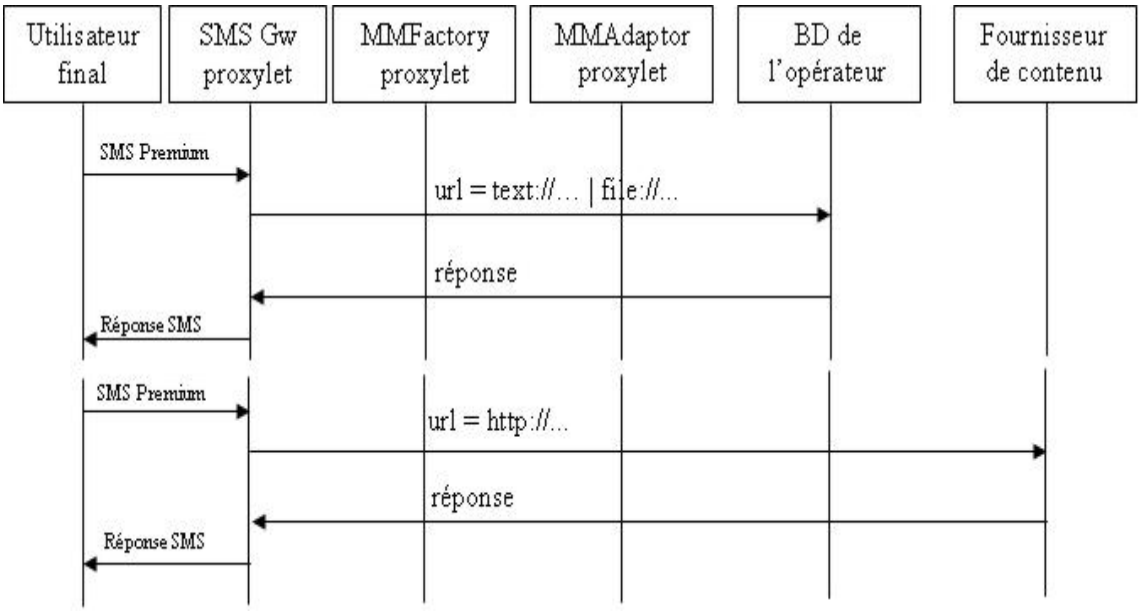


FIG. 5.3 – Envoi d'un SMS Premium.

L'utilisateur final envoie un SMS Premium à destination de la SMS Gateway qui, sur base du contenu du message, cherche le *mapping* correspondant et envoie la réponse à l'utilisateur final.

Rappelons que, comme le souligne la figure 5.3, le contenu de la réponse peut se trouver chez l'opérateur ou sur Internet (chez le fournisseur de contenu). Il sera dans ce cas récupéré à l'aide d'une requête HTTP.

Nous sommes ici confrontés au cas le plus simple, celui réalisable avec la SMS Gateway seule. La difficulté réside dans le fait qu'il ne peut y avoir interférence avec l'agent HTTP. Ce n'est pas le cas puisque l'URL de la requête HTTP ne possède pas le format nécessaire au traitement du flux par le *MMFactory*. Dès lors, le header "*X-mms_response*" n'est pas positionné et la proxylet *MMAdaptor* ne traite pas le flux de réponse.

5.2.2 Envoi d'un SMS Premium et réception d'un MMS (cas d'un *MM1 In Response*)

L'utilisateur final envoie un SMS Premium à destination de la SMS Gateway qui, sur base du contenu du message, cherche le *mapping* correspondant et envoie une requête HTTP formatée correctement pour déclencher la proxylet *MMFactory*.

La proxylet *MMFactory*, qui a accepté de traiter le flux, place le header "*X-mms_response*" et récupère les deux paramètres de l'URL :

1. provider (le nom du fournisseur de contenu).

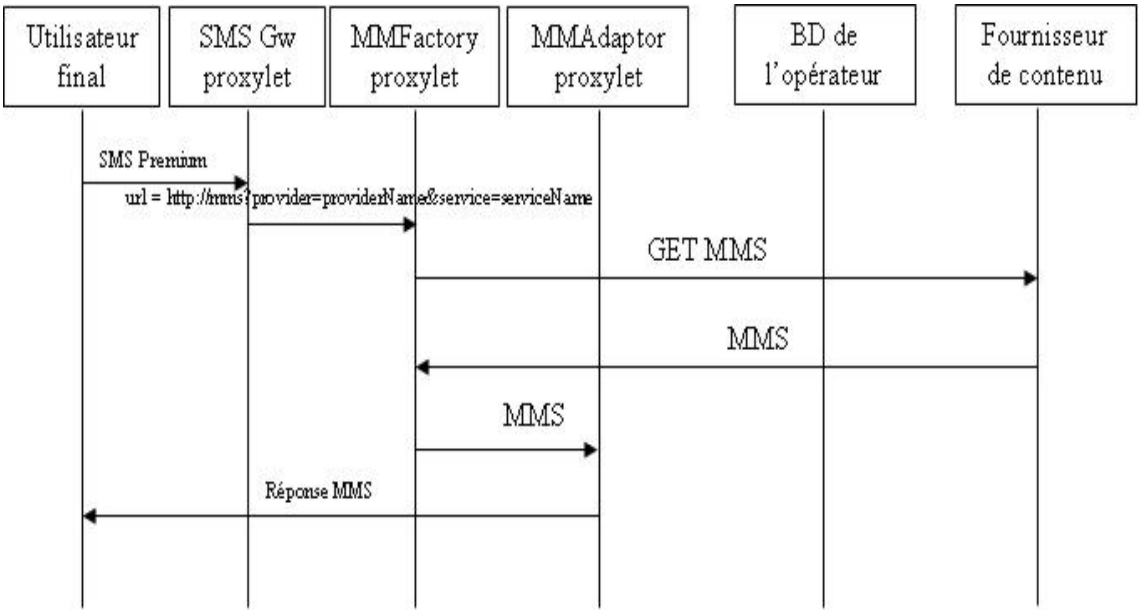


FIG. 5.4 – Envoi d'un SMS Premium, cas d'un ***MM1 In Response***.

2. service (le nom du service).

Ces deux paramètres lui permettent de charger le compte du fournisseur et le service concerné par le *mapping*.

Ensuite la proxylet vérifie le type de service. Il s'agit ici d'un service de type ***MM1 In Response***. Elle doit donc récupérer une URL indiquant où trouver le message MM1 qu'elle téléchargera via le protocole HTTP.

Une fois le message récupéré au format MM1, la *MMFactory* court-circuite la chaîne de requête pour laisser la main à la proxylet *MMAdaptor*. Celle-ci accepte le flux grâce au header "*X-mms_response*", encode le message au format MM7 et envoie le MMS à l'utilisateur final.

Mais comment la proxylet *MMAdaptor* sait-elle où envoyer le MMS ?

En fait, lorsque la SMS Gateway envoie une requête HTTP, elle positionne une série de header dont le header "*X-sms_from*" qui contient le numéro de téléphone de l'utilisateur final.

5.2.3 Envoi d'un SMS Premium et réception d'un MMS (cas d'un *MMSFactory*)

Le scénario est le même que précédemment.

C'est lors de la vérification du type qu'intervient la différence. Celui-ci n'est plus ***MM1 In Response***, mais ***MMSFactory***.

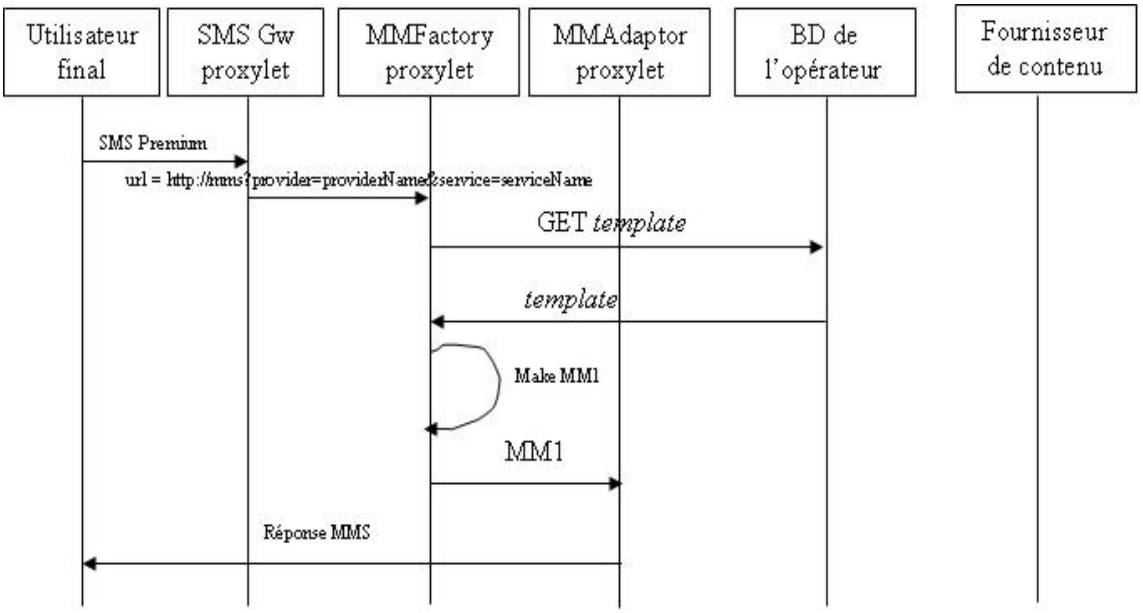


FIG. 5.5 – Envoi d'un SMS Premium, cas d'un ***MMSFactory***.

La *MMFactory* ne télécharge pas de MMS préfabriqué chez le fournisseur de contenu, mais récupère le *template* d'un MMS chez l'opérateur. A partir de ce *template*, elle construit un message au format MM1.

Enfin, la proxylet *MMAdaptor* récupère le message, l'encode au format MM7 et l'envoie à l'utilisateur final.

5.3 Extension de la SMS/MMS Gateway

Dans cette section, nous allons ajouter une fonctionnalité à la SMS/MMS Gateway. Bien que non implémentée lors de notre stage, cette fonctionnalité sera utile lors de la dernière partie de ce document.

La SMS/MMS Gateway telle que présentée jusqu'à présent, prend en entrée un SMS pour renvoyer en sortie un SMS ou un MMS. Il est possible, grâce à la modularité inhérente aux proxylets, d'étendre ces possibilités et faire en sorte qu'elle puisse prendre un MMS en entrée, sans pour autant bouleverser l'architecture de départ. Au lieu d'envoyer un SMS avec un ou plusieurs mots-clés, l'utilisateur final pourra envoyer un MMS contenant uniquement du texte. Ces MMS utiliseront l'interface MM7.

La solution la plus élégante serait de profiter des algorithmes, assez complexes, de gestion des *mappings*. Cela éviterait toute duplication de code. Ces algorithmes se situent dans la SMS Gateway. Or, du point de vue de la *Proxy Platform*, la SMS Gateway est un SMS agent. Il lui est donc impossible de recevoir un MMS qui est transporté par le protocole HTTP. Nous sommes confronté à un problème d'incompatibilité de protocoles.

Nous savons que la *Proxy Platform* peut nous être très utile pour ce genre de problèmes.

Nous allons placer une proxylet (*MM7 Filter*) supplémentaire dans la chaîne de requête de l'agent HTTP. Celle-ci filtrera les messages au format MM7 , récupérera le texte présent dans ces messages et l'enverra à l'agent SMS sous la forme d'un SMS.

Les interactions MM7 sont présentées à la figure 5.6. Il s'agit du scénario inverse de la figure 5.1. La SMS/MMS Gateway joue à nouveau le rôle du VAS et peut être considérée comme une application VAS à part entière. Une requête *MM7_ deliver.REQ* lui est envoyée par le MMSC.

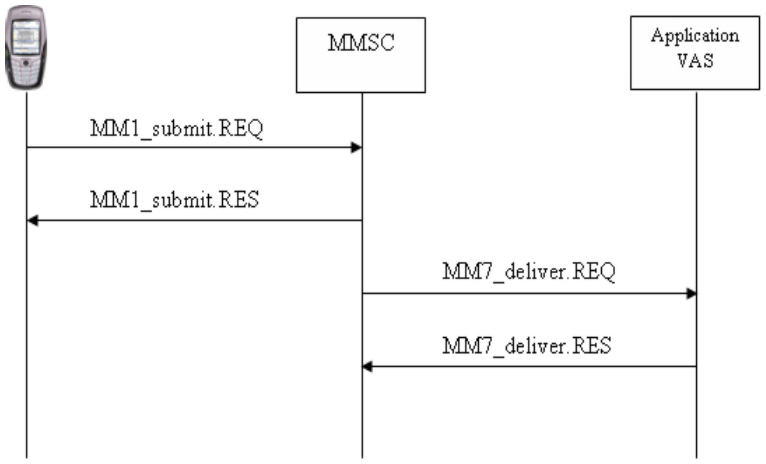


FIG. 5.6 – L'utilisateur final envoi un MMS (inspiré de [3, page 246]).

Le diagramme de séquence de la figure 5.7 nous montre les interactions au sein de la SMS/MMS Gateway.

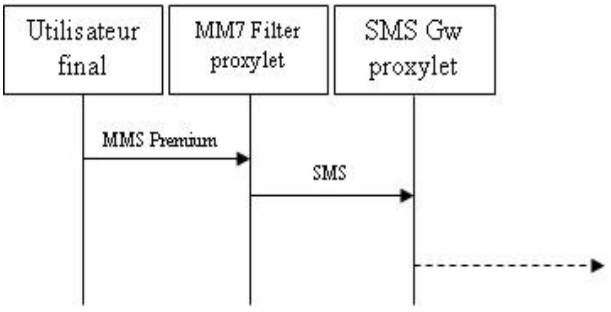


FIG. 5.7 – Envoi d'un MMS Premium.

L'utilisateur final envoie un MMS Premium. Ce MMS est intercepté par la proxylet *MM7 Filter* de l'agent HTTP. Celle-ci va récupérer le texte du MMS et envoyer un SMS à la proxylet SMS Gateway. Dès ce moment, nous revenons à la situation de base où l'utilisateur final envoie un SMS. Tous les scénarii décrits précédemment sont envisageables.

Grâce à cette modification 4 combinaisons sont possibles :

1. Envoi d'un SMS et réception d'un SMS.
2. Envoi d'un SMS et réception d'un MMS.

3. Envoi d'un MMS et réception d'un SMS.
4. Envoi d'un MMS et réception d'un MMS.

La figure 5.8 nous montre la nouvelle version de l'architecture. Comme nous l'avons précisé plus haut, il est relativement simple d'ajouter une telle fonctionnalité. Il suffit d'ajouter une proxylet (*MM7 Filter*) dans la chaîne de requête.

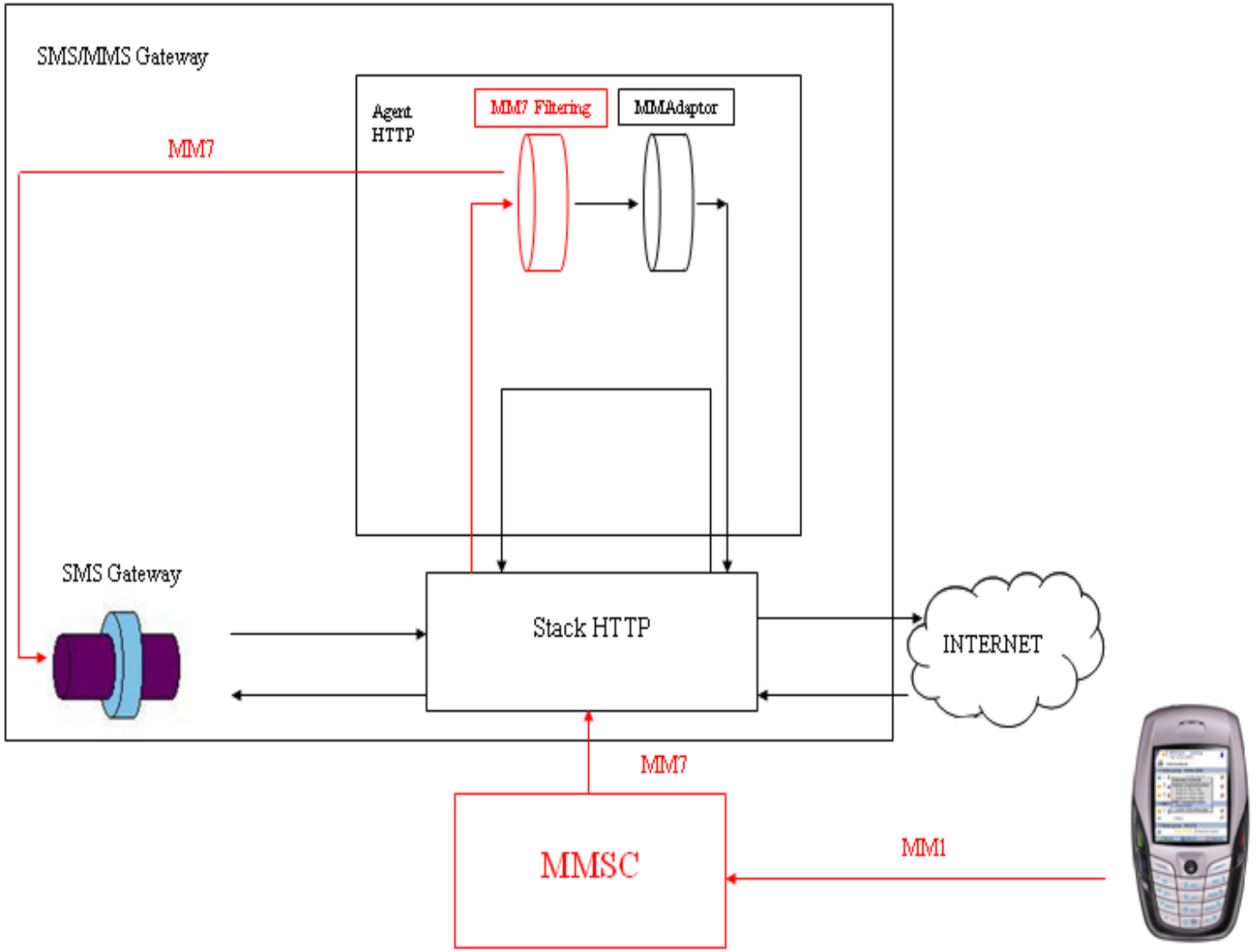


FIG. 5.8 – Nouvelle architecture de la SMS/MMS Gateway. Pour ne pas surcharger le schéma, nous n'avons pas représenté l'interface web ni les "comptes de fournisseurs".

5.4 Critique de la solution

Après avoir présenté la SMS/MMS Gateway telle qu'elle a été implémentée, nous allons prendre du recul et critiquer la solution. En effet, les contraintes commerciales auxquels sont liées les sociétés tels que *Nextenso S.A* nous limitaient dans la conception de la passerelle.

La démarche que nous allons utiliser s'éloignera de la logique d'une entreprise. Nous mettrons de côté la dimension "nombre de personnes" et temps nécessaire à la réalisation. Dès lors, nous oublierons la contrainte qui nous empêchait de modifier la SMS Gateway. Nous garderons cependant le même outil, à savoir, la *Proxy Platform*.

5.4.1 Une solution plus intégrée

La notion de "compte de fournisseur" n'était pas présente dans la SMS Gateway. Les services étaient apparentés à des *mappings* et ils n'appartenaient à priori à aucun fournisseur de contenu. Avec la solution que nous venons de présenter, la notion de "compte de fournisseur" n'existe que pour les MMS Premium. Or, un fournisseur de contenu peut offrir des services SMS Premium et MMS Premium.

La SMS/MMS Gateway telle que nous l'avons implémentée réagit différemment pour un service SMS Premium que pour un service MMS Premium :

- lorsqu'il s'agit d'un service MMS Premium, sur base du *mapping* lié au mot-clé reçu, notre passerelle charge le compte du fournisseur de contenu. A partir de ce compte et toujours sur base du *mapping*, elle récupère le service adéquat qui lui permet de générer la réponse ;
- lorsqu'il s'agit d'un service SMS Premium, sur base du *mapping* lié au mot-clé reçu, notre passerelle ne charge aucun compte, elle génère directement la réponse.

Autrement dit, si un fournisseur de contenu dispose d'un (ou plusieurs) service(s) MMS Premium, il est connu du système. Par contre s'il ne dispose que d'un (ou plusieurs) service(s) SMS Premium, il ne l'est pas.

Ceci est dû au fait que nous n'avons pas pu modifier la SMS Gateway. Dès lors, nous n'avons pas su y intégrer la notion de "compte de fournisseur"de contenu.

Nous pensons qu'il serait plus cohérent qu'un *mapping* soit associé à un "compte de fournisseur"qu'il s'agisse d'un service SMS Premium ou MMS Premium.

Après avoir présenté une première passerelle permettant de passer du SMS au MMS, nous allons analyser la faisabilité d'une passerelle permettant d'intégrer le SMS et le MMS dans le contexte plus large des IMS.

Troisième partie

PoC/MMS Gateway

Chapitre 6

Enregistrement d'un utilisateur non-IMS

Dans cette partie, nous allons analyser la faisabilité d'une passerelle qui a pour but d'intégrer les services actuels que sont le SMS et le MMS avec les services de demain, à savoir, les IMS. Nous la baptiserons PoC/MMS Gateway.

Ce genre de passerelle fait l'objet d'une demande de la part des opérateurs de téléphonie mobile. En effet, leur objectif à long terme est de faire en sorte que les IMS fassent partie intégrante de notre quotidien. Pour obtenir un tel résultat, ils devront s'en donner les moyens et offrir des solutions qui favorisent leur diffusion dans le grand public. La situation initiale verra un environnement où peu de personnes utiliseront les services IMS. Cette minorité sera isolée par rapport au monde non-IMS.

Cette situation est analogue à la diffusion de la vidéophonie. Les premiers terminaux permettant de faire de la vidéoconférence sont en train de faire leur apparition. Les utilisateurs de ce genre de technologie ne pourront en profiter énormément dans un premier temps, car, pour pratiquer la vidéophonie, il faut communiquer avec des interlocuteurs possédant des téléphones capables de la supporter. Comme ces terminaux font leur apparition, peu de personnes en possèdent. Dès lors, les utilisateurs seront, dans un premier temps, frustrés de ne pas pouvoir utiliser leur nouvel appareil.

Il faut donc offrir des solutions qui brisent la frontière entre les utilisateurs non-IMS et les utilisateurs IMS, favorisant ainsi la diffusion des IMS sans pour autant offrir aux utilisateurs non-IMS les facilités d'utilisation d'un utilisateur IMS. Sans quoi, ils ne seraient pas tentés de passer le cap.

La passerelle que nous allons présenter propose une de ces solutions. Elle a pour but de permettre à un utilisateur IMS de pouvoir communiquer via le Push To Talk avec un utilisateur non IMS, par l'intermédiaire du MMS.

Nous aborderons dans ce chapitre la problématique de l'enregistrement de l'utilisateur

non-IMS au système Push To Talk. Dans le chapitre suivant, nous proposerons une solution permettant à l'utilisateur non-IMS de recevoir un message en mode MULTIGROUP et en mode INVITE. Le chapitre 8 est dédié à l'étude du transfert de l'information dans l'autre direction : l'utilisateur non-IMS envoyant un MMS en réponse à un message Push To Talk initialement reçu. Enfin, nous montrerons, dans le dernier chapitre, une solution offrant à un utilisateur non-IMS la possibilité de modifier son information de présence.

Nous avons choisi délibérément d'utiliser la technologie Proxy Platform présentée au chapitre 3. Bien que d'autres types de technologie existent, celle-ci s'est imposée grâce à ses nombreuses qualités en matière de développement d'applications réseau et notamment par le fait qu'elle permet de résoudre les problèmes d'incompatibilité liés au protocole. De plus, il s'agit d'une technologie que nous maîtrisons pour l'avoir étudiée et mise en pratique.

Comme nous l'avons vu dans le chapitre 2, l'enregistrement a pour but d'associer une adresse IP à une SIP URI. Dans le cas d'un utilisateur non-IMS, nous associerons à sa SIP URI l'adresse IP de la PoC/MMS Gateway.

Il faut trouver un moyen de permettre à un utilisateur non-IMS de s'inscrire au système Push To Talk. Nous ne pensons pas qu'il soit pratique pour l'utilisateur de se connecter au système dès qu'il se connecte au réseau de mobile (c'est-à-dire lorsqu'il démarre son téléphone portable). En effet, tout comme un utilisateur IMS, l'utilisateur non-IMS peut avoir envie de connecter son terminal au réseau sans pour autant recevoir des messages Push To Talk. Nous devons lui laisser le choix.

L'enregistrement doit se faire à la suite d'une démarche de l'utilisateur. Une solution très simple serait l'envoi d'un SMS ou d'un MMS. L'utilisateur aurait la possibilité d'envoyer un message contenant le mot-clé **REGISTER** au numéro court 3310 par exemple. Ceci nous fait étrangement penser aux services SMS et MMS Premium traités lors de la première partie. Nous allons profiter de ce qui existe déjà pour réaliser notre solution. Nous opterons pour le SMS Premium pour deux raisons :

1. Le coût d'un SMS est beaucoup moins élevé que celui d'un MMS.
2. Le format du SMS (160 caractères) est tout à fait adapté et suffisant pour l'envoi de tels messages.

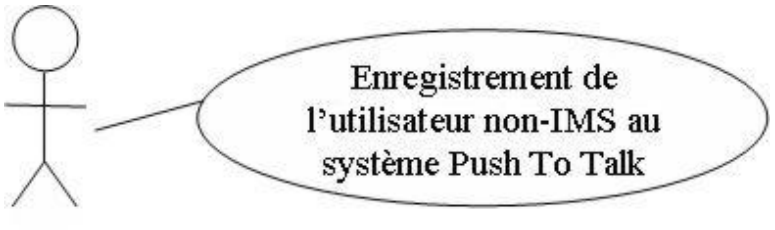
Une fois le SMS envoyé, l'utilisateur sera enregistré auprès du système Push To Talk. Ce qui signifie que toutes les requêtes SIP qui lui seront destinées seront redirigées vers la PoC/MMS Gateway. Comme nous le montrerons plus loin, il apparaîtra également dans la liste de contacts de tous les contacts auprès desquels il s'est enregistré. Lorsqu'une personne voudra lui envoyer un message Push To Talk, celle-ci le sélectionnera dans sa liste, appuiera sur le bouton adéquat pour parler. Le message lui sera alors envoyé sous forme d'un MMS. Cette phase est donc essentielle avant toute interaction.

Le diagramme de séquence décrivant l'enregistrement au système Push To Talk est décrit à la figure 2.2 (chapitre 2). L'interaction qui nous intéresse ici se trouve en amont

de ce schéma. Nous allons l'analyser.

Nous utiliserons le même canevas tout au long de cette partie. Nous commencerons par donner un use case formel du scénario étudié. Ensuite nous présenterons les diagrammes de séquence les plus intéressants montrant les interactions entre les différents composants. A la fin de ce chapitre, nous montrerons l'architecture issue de notre analyse.

6.1 Use case



Pré Condition :

- l'utilisateur non-IMS possède un mobile supportant le SMS ;
- l'utilisateur non-IMS n'est pas connecté au système Push To Talk.

Post Condition :

- l'utilisateur est enregistré au système Push To Talk. Il peut désormais recevoir des messages Push To Talk sous la forme de MMS.

Scénario :

Utilisateur	**PoC/MMS Gateway**
1. L'utilisateur non -IMS envoie un SMS contenant un mot-clé à un numéro court.	
	2. La PoC/MMS Gateway reçoit une requête HTTP en provenance du système et enregistre l'utilisateur à l'origine de la requête.
3. L'utilisateur non-IMS reçoit un SMS lui confirmant qu'il est bien enregistré au système Push To Talk.	

6.2 Diagrammes de séquence

Ce scénario utilise trois protocoles différents qui sont : SMS, HTTP, SIP.

Le diagramme de séquence de la figure 6.1 illustre les interactions entre les différents acteurs.

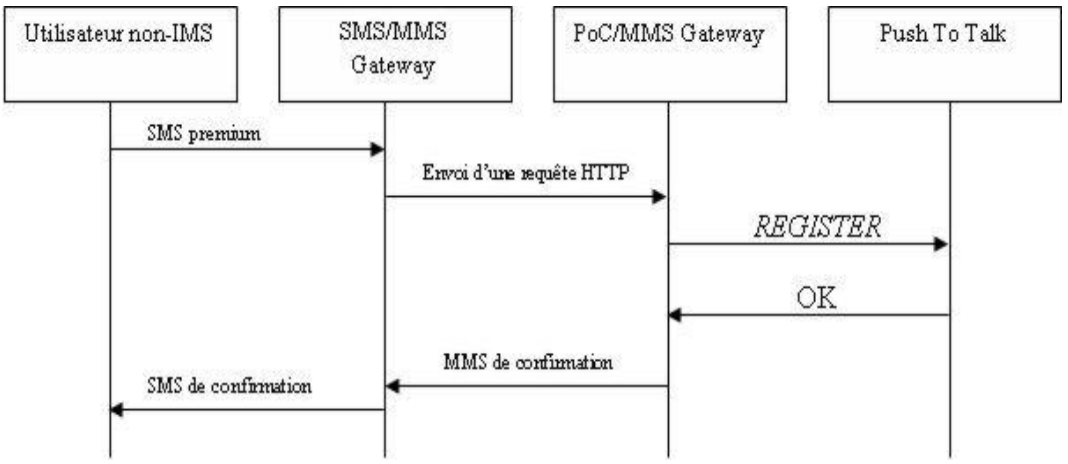


FIG. 6.1 – Processus d'enregistrement d'un utilisateur au système Push To Talk.

Partons de la PoC/MMS Gateway et posons-nous la question de savoir quelle information elle requiert pour enregistrer un utilisateur.

Seule la SIP URI de l'utilisateur non-IMS est utile pour cette opération. Il faut fournir à la PoC/MMS Gateway une SIP URI en partant d'un SMS qui sera envoyé par l'utilisateur non-IMS. Un moyen très simple pour y arriver consiste à envoyer un SMS Premium contenant un mot-clé accompagné de la SIP URI de l'utilisateur (par exemple : "**REGISTER** *sip :+32476865596@proximus.be*").

Nous allons pouvoir profiter des *mappings* de type "*expression régulière*" pour configurer notre SMS/MMS Gateway. De cette façon, nous devrons configurer un seul *mapping* pour tous les utilisateurs non-IMS souhaitant s'enregistrer au système Push To Talk. Par conséquent, si l'on prend le mot-clé **REGISTER**, nous devrons définir l'expression régulière suivante :

REGISTER *\s+(\S+)$*

Lorsque la SMS/MMS Gateway recevra un tel SMS, elle enverra une requête HTTP en direction de la PoC/MMS Gateway. Nous utiliserons le même mécanisme que celui utilisé pour étendre la SMS Gateway, à savoir le recours à une requête HTTP spécifique contenant le ou les paramètres qui nous seront utiles. Dans le cas présent, un seul paramètre est nécessaire, celui contenant la SIP URI de l'utilisateur non-IMS souhaitant s'enregistrer. Les requêtes HTTP qui concernent la PoC/MMS Gateway pourraient avoir l'allure suivante :

```
http://ptt?sipuri=SipUri
```

Comme nous pouvons le voir sur la figure 6.2, tout comme l'agent HTTP étendant la SMS Gateway était proxy de celle-ci, nous placerons la PoC/MMS Gateway comme proxy

de la SMS/MMS Gateway. La PoC/MMS Gateway ne traitera que les requêtes qui lui seront destinées, c'est-à-dire celles qui ont le format adéquat.

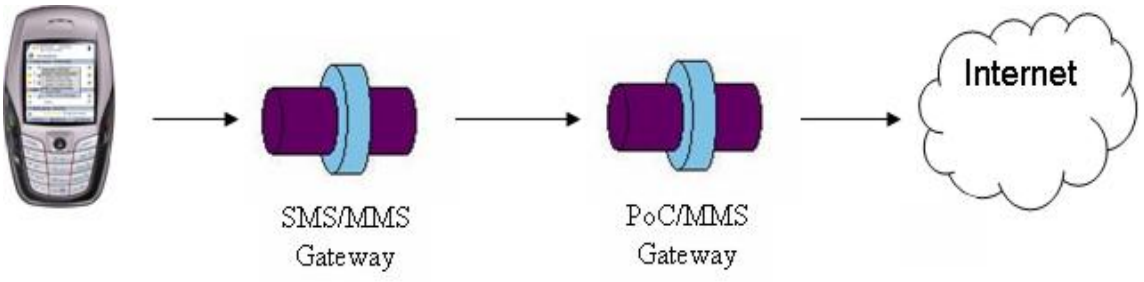


FIG. 6.2 – Enchaînement de proxies.

Intéressons nous maintenant de plus près à ce qui se passe au sein de la PoC/MMS Gateway. La figure 6.3 nous montre le diagramme de séquence illustrant les interactions entre les différents éléments lors d'un enregistrement.

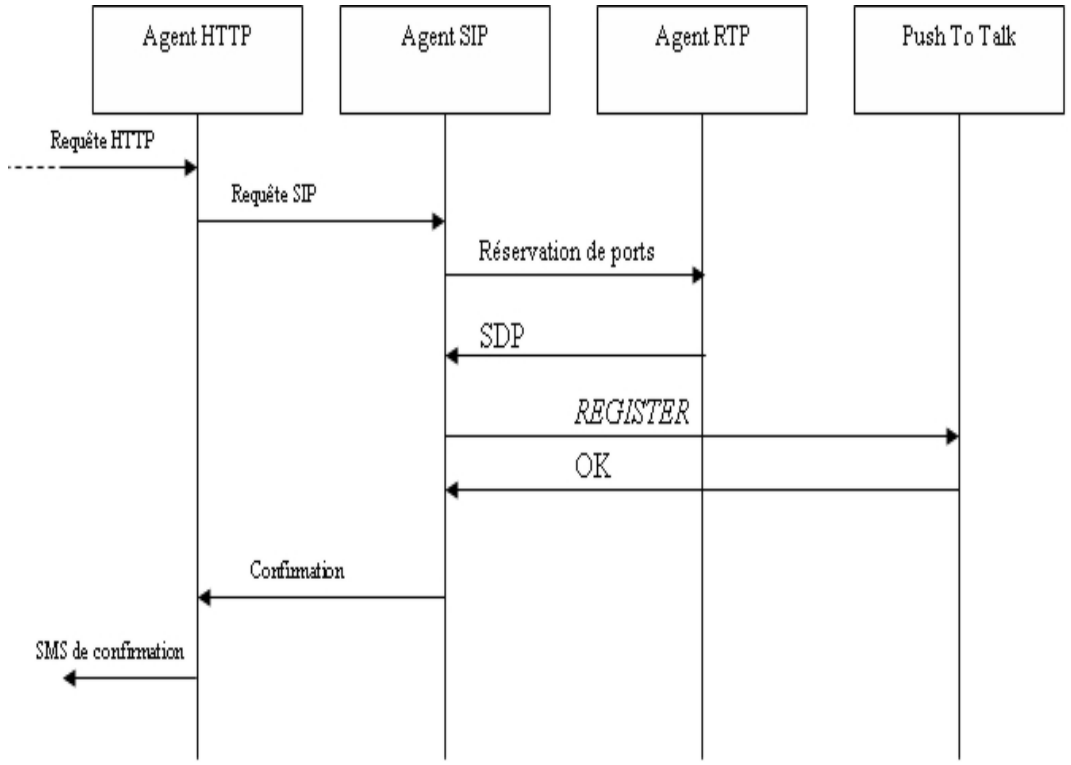


FIG. 6.3 – Interactions au sein de la PoC/MMS Gateway.

Pour cette opération, notre passerelle prend en entrée une requête HTTP pour ressortir une requête SIP *REGISTER* contenant un document SDP. Un agent HTTP est utile pour recevoir la requête HTTP alors qu'un agent SIP est utile pour envoyer la requête *REGISTER* au système Push To Talk. Il faut trouver un moyen de faire communiquer ces deux agents. Une solution est de placer au sein l'agent HTTP un client SIP qui aura pour seule fonction de faire suivre les requêtes à l'agent SIP. Nous utilisons une requête SIP adaptée,

exactement comme pour les requêtes HTTP en provenance de la SMS/MMS Gateway et à destination de l'agent HTTP de notre PoC/MMS Gateway. Enfin, nous aurons besoin de gérer les interactions RTP/RTCP. Pour ce faire, un agent RTP s'avère utile. Lors de la phase d'enregistrement, il est nécessaire de fournir un document SDP à l'application Push To Talk ; il permettra l'établissement d'une session multimédia. Il contient notamment les ports sur lesquels se connecter. Ce document sera fourni par l'agent RTP sur base d'une requête émanant de l'agent SIP. L'agent RTP sera chargé de réserver des ports pour l'utilisateur qui s'enregistre. Deux ports sont nécessaires, un port RTP et un port RTCP.

Précisons que nous utilisons le protocole RTP au-dessus du protocole UDP. Par conséquent nous parlons de réservation de ports et non d'allocation de ports. En effet, contrairement à TCP, aucune connexion n'est établie entre un émetteur et un récepteur. Il n'y a dès lors aucune consommation de ressources[1].

L'utilisateur non-IMS doit recevoir un SMS lui confirmant qu'il est bien enregistré au système. Dès lors, lorsque le client SIP de l'agent HTTP reçoit la réponse, il doit envoyer un SMS à l'utilisateur non-IMS. Or, le protocole HTTP ne convient pas à l'envoi de SMS. Par contre il permet l'envoi de MMS par l'interface MM7. Nous allons ainsi profiter de la dernière fonctionnalité étudiée dans la deuxième partie de ce document. Nous avions étendu les possibilités de la SMS/MMS Gateway en lui permettant de prendre en entrée un MMS et sortir un SMS. Nous passerons par l'interface MM7 pour envoyer un message contenant un mot-clé, auquel sera associé un texte statique, qui contiendra le message prédéfini. Ce message sera invariable en fonction de la personne, nous pouvons par conséquent définir un *mapping* "*mot-clé*" auquel sera associé la réponse du message.

Mais comment la SMS/MMS Gateway va-t-elle savoir à qui envoyer le message ?

En effet, la SMS/MMS Gateway renvoie toujours ses messages en réponse à un expéditeur initial. Ici, l'expéditeur est notre PoC/MMS Gateway ; il ne faut donc pas que le SMS soit redirigé vers celle-ci. Il faudra dès lors que l'agent HTTP place dans le header "*From*" du MMS, le numéro de téléphone de l'utilisateur final. Ce numéro se trouvera dans la SIP URI.

6.3 Architecture

La figure 6.4 montre l'architecture de la PoC/MMS Gateway. Comme nous pouvons le voir, celle-ci est composée de trois agents :

1. L'agent RTP est chargé de créer le document SDP à fournir au système Push To Talk.
2. Sur base d'un message au format MM7, l'agent HTTP fait suivre la requête à l'agent SIP. Il est également chargé d'envoyer la confirmation au format MM7 à la SMS/MMS

[1] Le lecteur pourra se reporter à [6] pour plus d'informations à ce sujet.

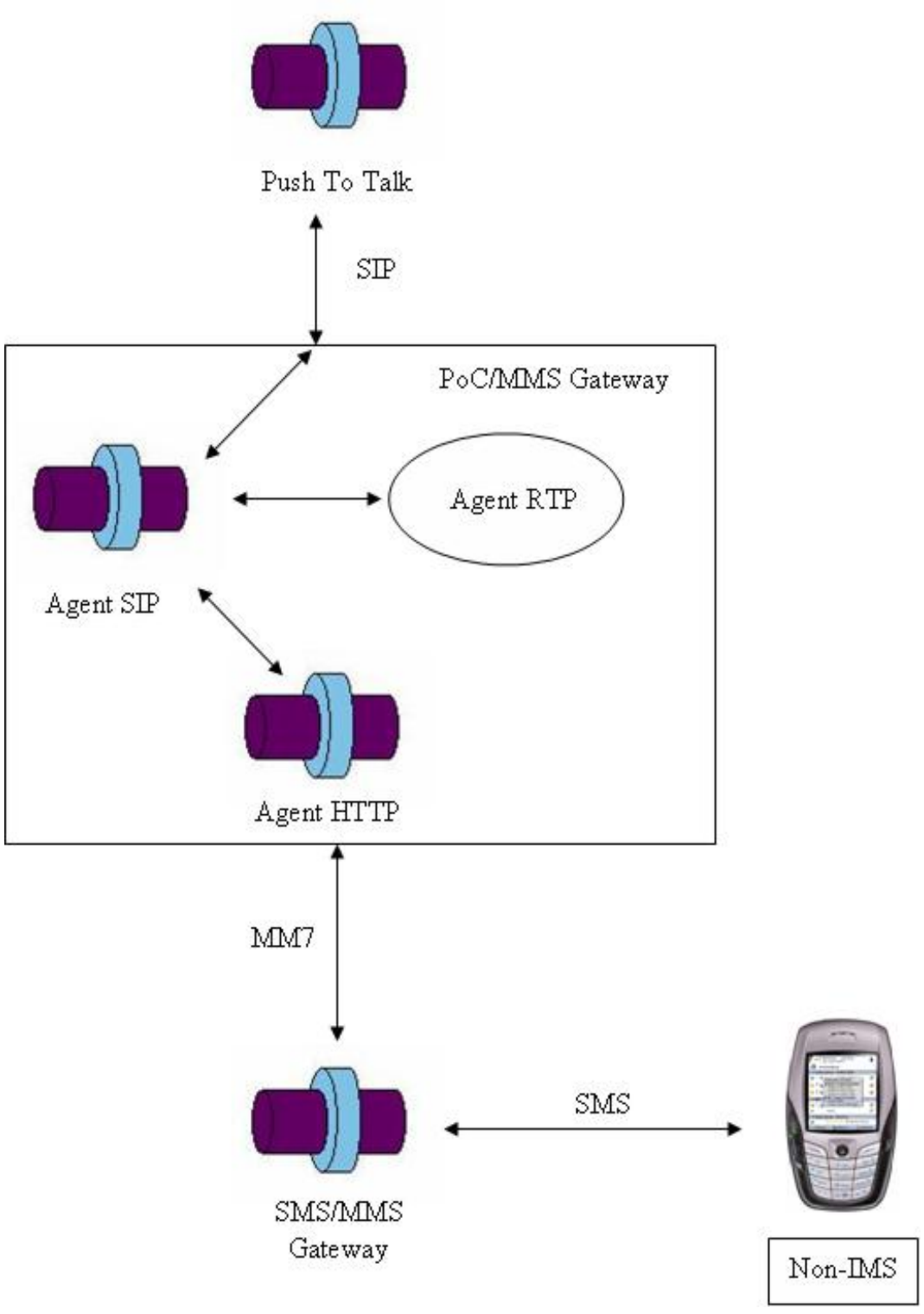


FIG. 6.4 – Architecture de la PoC/MMS Gateway.

Gateway.

3. L'agent SIP gère les interactions SIP avec le système Push To Talk.

Chapitre 7

Envoi d'un message Push To Talk

Après avoir présenté une solution permettant à un utilisateur non-IMS de se connecter au système Push To Talk, nous allons étudier la possibilité pour un utilisateur IMS de lui envoyer un message Push To Talk.

Pour qu'un utilisateur IMS puisse envoyer un message Push To Talk à un utilisateur non-IMS, l'utilisateur non-IMS doit apparaître dans la liste de contacts de l'utilisateur IMS. Nous reviendrons en détails sur ce point dans le chapitre 9.

Nous étudierons la possibilité d'envoyer un message Push To Talk dans deux types de conférences : le mode MULTIGROUP et le mode INVITE.

7.1 Envoi d'un message en mode MULTIGROUP

Le scénario est présenté à la figure 7.1. L'utilisateur IMS envoie un message Push To Talk par l'intermédiaire de son téléphone mobile à un utilisateur non-IMS. Ce message est dirigé vers la PoC/MMS Gateway qui l'encapsule dans un MMS. Celui-ci est finalement envoyé au destinataire.

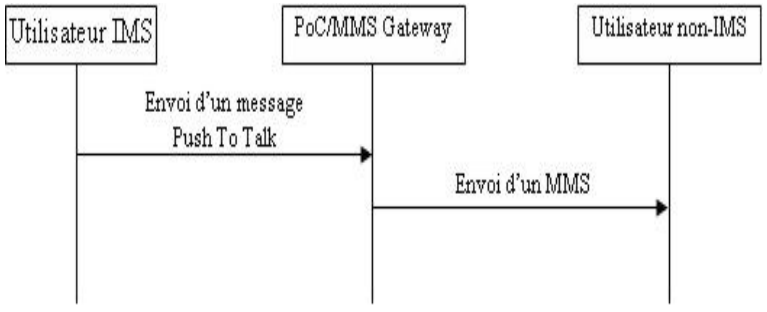


FIG. 7.1 – Vue générale du scénario.

Il est important de comprendre que le Push To Talk est un système de messagerie instantanée, ce qui n'est pas le cas du MMS. Par conséquent, la latence inhérente à l'uti-

lisation du MMS rend la nature même de celui-ci antinomique au concept de messagerie instantanée. La solution que nous présentons n'est qu'un palliatif visant à faire interopérer deux types de messagerie intrinsèquement opposés. Elle n'a pas la prétention de permettre à un utilisateur IMS de communiquer instantanément avec un utilisateur non-IMS. Nous aurons affaire à des conversations légèrement différées.

7.1.1 Use case

Pré Condition :
– l'utilisateur non-IMS est connecté au système Push To Talk ;
– l'utilisateur non-IMS possède un mobile supportant le MMS.
Post Condition :
– l'utilisateur non-IMS reçoit le message initial encapsulé dans un MMS.
Scénario :

Utilisateur	**PoC/MMS Gateway**
1. L'utilisateur IMS envoie un message Push To Talk à un ou plusieurs membres de sa liste de contacts. Ce message est envoyé à une ou plusieurs personnes non-IMS.	
	2. La PoC/MMS Gateway reçoit le message, l'encapsule dans un MMS et l'envoie aux utilisateurs non-IMS.

Nous précisons que l'utilisateur non-IMS doit posséder un mobile supportant le MMS. Cette condition n'est pas rédhibitoire pour le scénario lui-même. En effet, lorsqu'un utilisateur reçoit un MMS alors que son mobile ne le supporte pas, il peut recevoir un SMS lui indiquant une URL où aller consulter ce MMS. Cette situation amplifie fortement le phénomène de latence lié au MMS. De plus, il sera impossible à l'utilisateur non-IMS de répondre au message reçu. La conversation sera dans ce cas unidirectionnelle.

7.1.2 Diagrammes de séquence

Pour rappel, lors d'une conférence en mode MULTIGROUP, un utilisateur sélectionne un ou plusieurs membres de sa liste de contacts, appuie sur un bouton et parle. Ce message est alors envoyé en *streaming*.

Au préalable, tout utilisateur a dû envoyer, au moment de l'enregistrement, un message *INVITE* au système contenant la description de la session (cfr chapitre 2). Nous devons par conséquent compléter notre phase d'enregistrement illustrée à la figure 6.3 du chapitre 6. La figure 7.2 nous montre la nouvelle mouture de la phase d'enregistrement d'un utilisateur non-IMS au système Push To Talk. Seule une requête *INVITE* est ajoutée après l'envoi de la requête *REGISTER*.

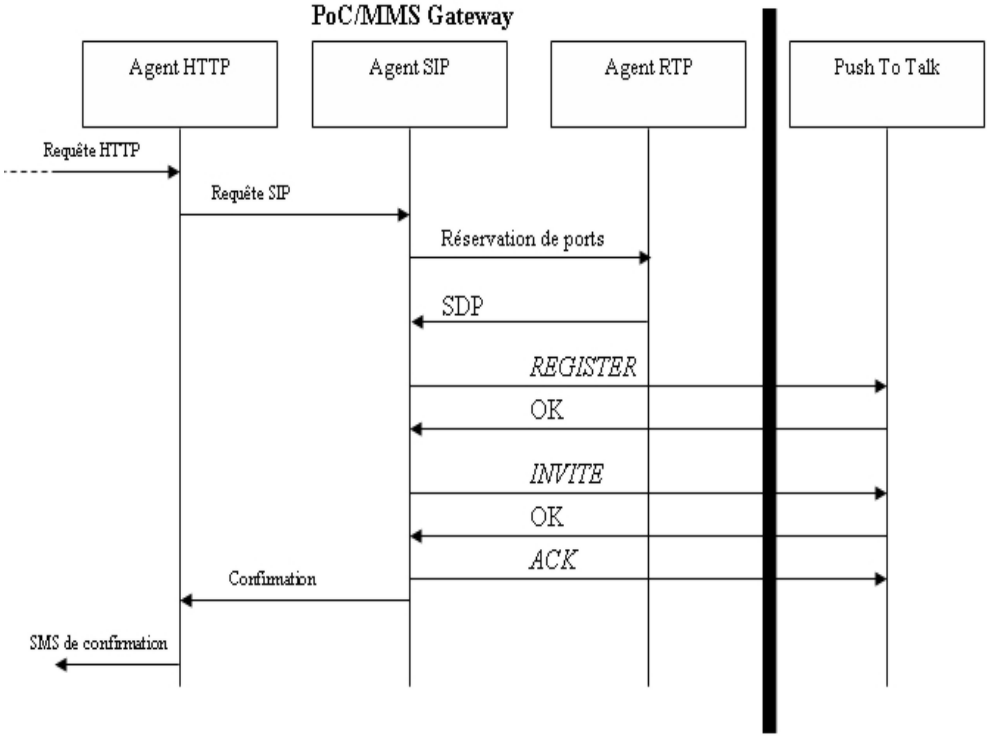


FIG. 7.2 – Nouvelle version de la phase d'enregistrement.

Attardons-nous maintenant à l'interaction RTP/RTCP qui est illustrée à la figure 7.3.

Le *Broadcaster* RTP de l'application Push To Talk envoie les messages à destination d'utilisateurs non-IMS vers la PoC/MMS Gateway. Le message vocal est intercalé entre un *SOS* (Start Of Speech) indiquant le début d'un message et un *EOS* (End Of Speech) indiquant la fin de ce message. Lorsque la PoC/MMS Gateway reçoit le message *EOS*, elle encapsule le message vocal dans un MMS et l'envoie à l'utilisateur non-IMS par l'interface MM7.

Analysons de plus près comment notre passerelle pourrait gérer une telle situation. Dans ce scénario, elle prend en entrée un message RTP et produit en sortie un MMS via

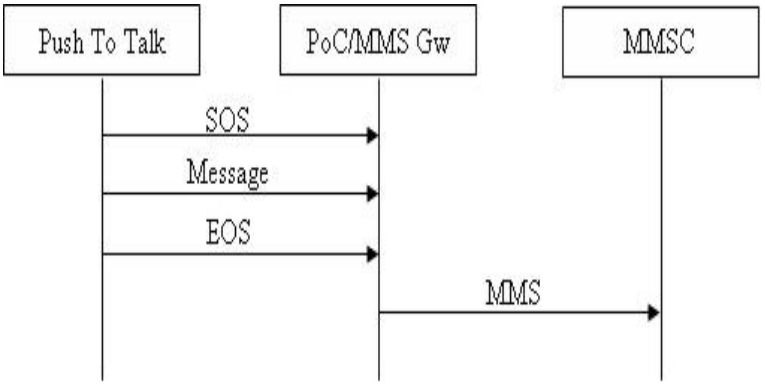


FIG. 7.3 – Envoi d'un message RTP.

l'interface MM7, c'est-à-dire une requête HTTP.

L'agent RTP intercepte les différents messages RTP/RTCP. A chaque réception d'un *EOS*, il envoie une requête HTTP à l'agent HTTP. Le corps de cette requête contiendra le message vocal. Il sera nécessaire de compresser le message au maximum afin de ne pas encombrer le réseau. Nous ne nous attarderons cependant pas sur l'aspect compression de données qui sort du cadre du présent document. Lorsque l'agent HTTP reçoit la requête HTTP émanant de l'agent RTP, il construit un MMS et l'envoie par l'interface MM7.

Mais comment l'agent HTTP sait-il à quel numéro envoyer le message ?

C'est en fait l'agent RTP qui connaît ce numéro. Comme nous l'avons dit plus haut, avant l'interaction vocale, une requête SIP *INVITE* a été envoyée. Celle-ci contient un document SDP décrivant la session multimédia (notamment l'adresse IP et le port). Lorsque la PoC/MMS Gateway reçoit un tel message, elle réserve deux ports (un port RTP et un port RTCP) à l'utilisateur. L'agent RTP peut donc associer à chaque port le numéro de téléphone d'une personne.

L'agent RTP qui connaît le numéro de téléphone du destinataire doit pouvoir le transmettre à l'agent HTTP. Nous avons choisi de placer ce numéro dans un header de la requête HTTP. Lorsque l'agent HTTP recevra la requête, il récupèrera le numéro de téléphone du destinataire dans ce header et pourra ainsi envoyer le MMS à la bonne personne.

Le diagramme complet de l'interaction se trouve à la figure 7.4.

7.2 Envoi d'un message en mode INVITE

Dans une conférence en mode INVITE, plusieurs personnes sont en conversation autour d'un thème. Dans ce genre de conversation, une seule personne peut prendre la parole. Pour ce faire, elle doit appuyer sur un bouton de son téléphone jusqu'à l'obtention du "*droit de parler*". Une fois ce droit acquis, elle envoie son message vocal qui est diffusé à tous les participants à la conversation. Les personnes désireuses de réagir doivent procéder de la

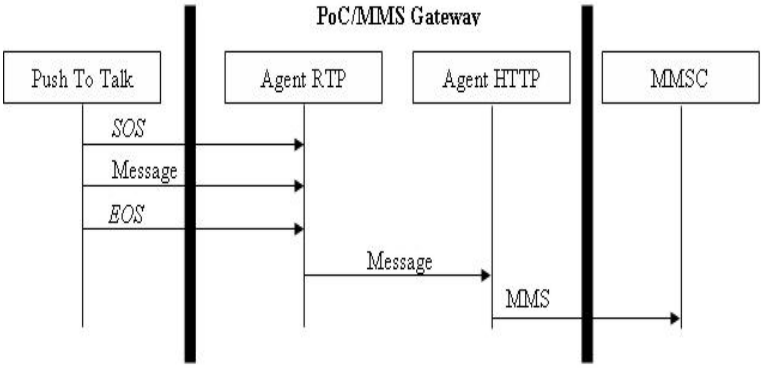


FIG. 7.4 – Interactions au sein de la PoC/MMS Gateway.

même façon.

Dans un tel type de conversation, notre passerelle n'apparaît pas extrêmement pratique. La latence liée au MMS fait que l'utilisateur non-IMS ne sera jamais en phase dans la conversation. Le temps écoulé entre le moment où une personne émet un message, le moment où l'utilisateur non-IMS récupère le MMS et le moment où la réponse de cet utilisateur est diffusée dans la conversation sera beaucoup trop important (de l'ordre de plusieurs dizaines de secondes). Dans de telles conditions, il n'est pas réaliste de faire interagir un utilisateur non-IMS dans ce genre de conversation.

Nous pouvons cependant imaginer un scénario qui permettrait d'utiliser la passerelle dans une conversation en mode INVITE. Celle-ci pourrait servir d'enregistreur de conversation. Au lieu de recevoir les messages un par un sans pouvoir réagir, l'utilisateur recevrait un MMS contenant l'enregistrement complet de la conversation qui aurait eu lieu entre les utilisateurs IMS. Cette façon de faire peut avoir un effet frustrant pour l'utilisateur non-IMS.

Imaginons que l'utilisateur A invite l'utilisateur B et l'utilisateur C dans une conversation en mode INVITE. A et B sont des utilisateurs IMS tandis que C est non-IMS. Seuls A et B vont interagir. C, lui, ne recevra que l'enregistrement de la conversation entre A et B. Supposons que A et B débattent du film qu'ils vont aller voir au cinéma le soir. Cette conversation aboutira sur un rendez-vous à un lieu et une heure précise, pour aller voir un film bien précis. C, lui, recevra l'enregistrement de la conversation et n'aura par conséquent pas pu donner son avis.

Ce que nous voulons montrer par ce petit exemple, c'est le sentiment de frustration que pourrait entraîner notre passerelle sur un utilisateur non-IMS dans pareille situation, poussant celui-ci à passer dans le monde IMS. Nous avions précisé plus haut que le but de la PoC/MMS Gateway était de briser les frontières qui peuvent exister entre les utilisateurs IMS et les utilisateurs non-IMS. Avec l'exemple que nous venons de présenter, nous montrons qu'il est également possible de pousser les utilisateurs non-IMS à devenir utilisateurs IMS. Ce genre de scénario est tout à fait favorable à la diffusion des IMS.

Nous allons maintenant définir formellement ce scénario et analyser comment il serait possible de l'implémenter.

7.2.1 Use Case

Pré Condition :

– l'utilisateur non-IMS est connecté au système Push To Talk.

Post Condition :

– l'utilisateur non-IMS reçoit un enregistrement de la conversation encapsulé dans un MMS.

Scénario :

Utilisateur	**PoC/MMS Gateway**
1. L'utilisateur IMS invite un ou plusieurs membres de sa liste de contacts. Au moins une des personnes invitées est non-IMS.	
	2. La PoC/MMS Gateway accepte l'invitation.
3. Les utilisateurs IMS interagissent	
	4. La PoC/MMS Gateway enregistre la conversation.
5. L'initiateur de la conversation termine celle-ci.	
	6. La PoC/MMS Gateway envoie l'enregistrement de la conversation aux utilisateurs non-IMS.

Notons que dans ce scénario, nous ne précisons pas que le mobile de l'utilisateur non-IMS doit supporter le MMS. Le principal inconvénient dans le cas d'une interaction en mode INVITE est le fait que l'utilisateur ne pourrait pas répondre au message. Or, ici, comme il est précisé que l'utilisateur non-IMS ne pourra pas interagir, on peut tout à fait

admettre qu'il reçoive un SMS contenant une URL où il pourra récupérer le message. De plus, la latence ne sera pas problématique ici.

7.2.2 Diagrammes de séquence

Nous allons traiter ici de l'interaction au niveau SIP et au niveau RTP.

Le diagramme de séquence de la figure 7.5 illustre l'interaction SIP entre les différents acteurs.

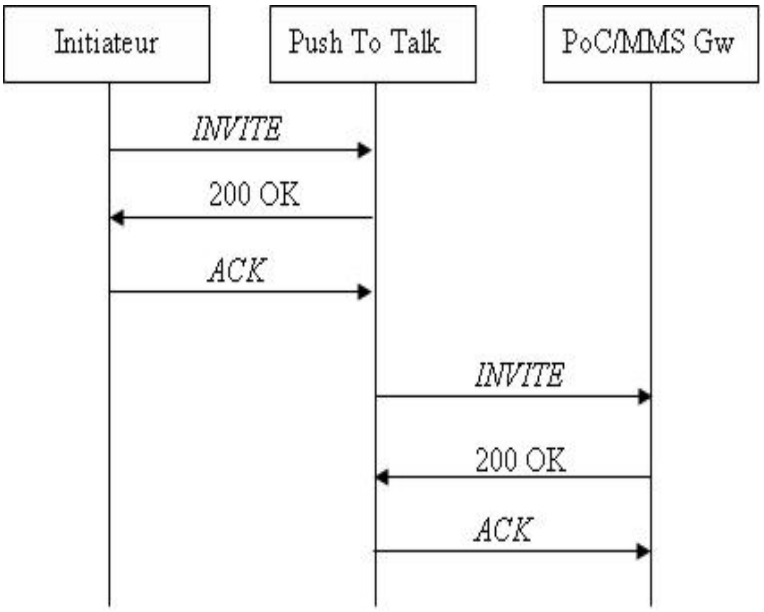


FIG. 7.5 – Initiation de la conférence.

L'initiateur de la conférence envoie un message *INVITE* au système contenant la liste des invités. Comme précisé dans le use case, cette liste contient au moins un utilisateur non-IMS. Lorsque le système Push To Talk reçoit un tel message, il envoie des messages *INVITE* secondaires à tous les membres de la liste des invités. Pour chaque utilisateur non-IMS présent dans la liste, le message *INVITE* secondaire sera envoyé à la PoC/MMS Gateway car leur SIP URI est associée à l'adresse IP de la passerelle. Pour simplifier le schéma, nous montrons uniquement les messages *INVITE* secondaires échangés entre le système Push To Talk et notre passerelle. Les autre messages secondaires ne nous intéressent pas.

En réponse au message *INVITE*, le système Push To Talk attend un message OK (s'il y a possibilité d'établir la connexion) contenant un document SDP. Pour rappel, ce document SDP donne les informations nécessaires à l'établissement d'une liaison RTP. Plus précisément, notre passerelle renverra les informations telles que l'adresse IP et les ports RTP/RTCP qu'elle aura réservés à l'utilisateur pour la conversation. Dorénavant, lorsque la PoC/MMS Gateway recevra des messages sur ce port, elle l'enregistrera.

L'enregistrement de la conversation se terminera lorsque l'initiateur terminera celle-ci,

c'est-à-dire lorsque la PoC/MMS Gateway recevra un message SIP *BYE* de la part de l'initiateur de la conversation.

Nous allons maintenant présenter en détails les interactions au sein même de notre passerelle pour un tel type de scénario. Elle prend en entrée un message SIP et produit en sortie un message SIP.

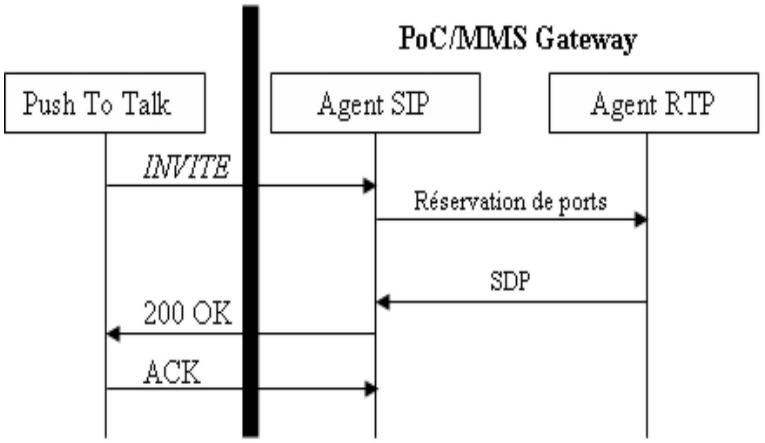


FIG. 7.6 – Interactions au sein de la PoC/MMS Gateway.

L'agent SIP reçoit une requête *INVITE* sur base de laquelle il envoie un message à l'agent RTP afin de réserver des ports pour une conversation (figure 7.6). Ce message contiendra le numéro de téléphone de l'expéditeur.

Comme nous l'avions précisé précédemment, lorsqu'une conversation est terminée, l'agent RTP la renvoie dans une requête HTTP à l'agent HTTP. Celui-ci doit encapsuler le message vocal dans un MMS et l'envoyer. Pour ce faire, il a besoin du numéro de téléphone de la personne pour qui le message est destiné. Le champ "*To*" du message *INVITE* contient la SIP URI du destinataire. Cette SIP URI contiendra le numéro de téléphone du destinataire :

```
sip:numero@domaine
```

Une fois le message reçu en provenance de l'agent SIP, l'agent RTP renverra le document SDP qui sera placé dans le corps de la réponse SIP au système Push To Talk.

Dès que tous les invités auront répondu à l'invitation de l'initiateur de la conférence, elle sera ouverte. Notre agent RTP sera à ce moment susceptible de recevoir des messages en provenance du système Push To Talk. L'interaction RTP sera similaire à celle que nous avons présenté en mode MULTIGROUP. Cependant, à la réception d'un *EOS*, l'agent RTP ne devra pas envoyer le message à l'agent HTTP mais l'ajouter à l'enregistrement de la conversation.

Ce n'est que lors de la réception du message SIP *BYE* que la procédure d'envoi du MMS sera possible.

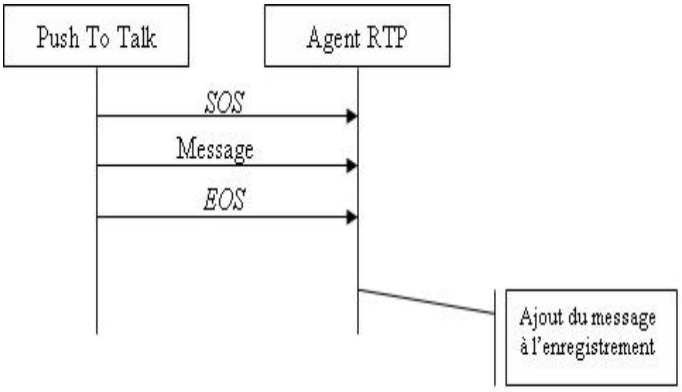


FIG. 7.7 – Envoi d'un message RTP.

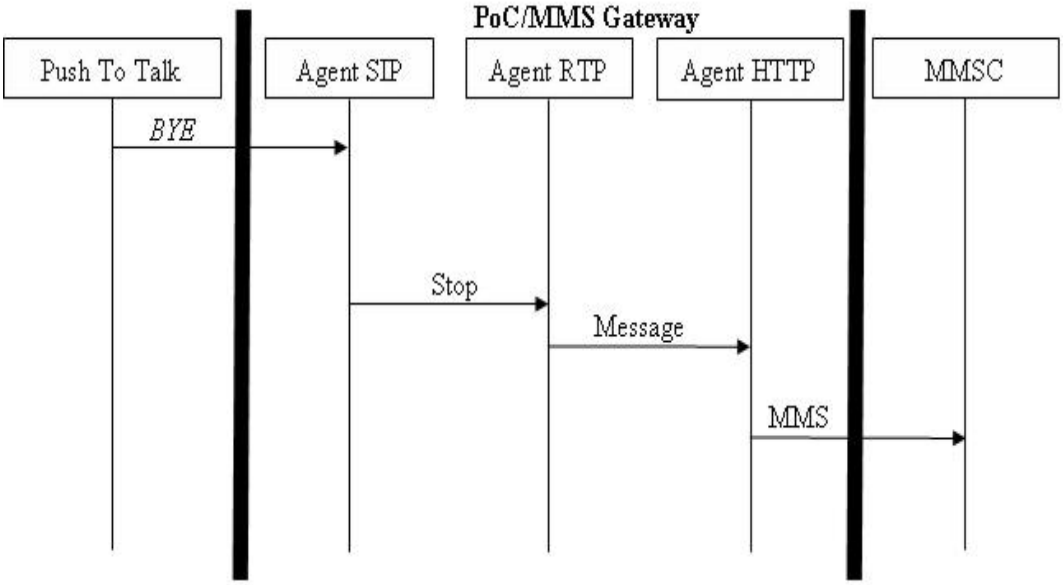


FIG. 7.8 – Interactions au sein de la PoC/MMS Gateway.

Lorsque l'initiateur de la conférence quitte celle-ci, il envoie un message SIP *BYE* à l'application Push To Talk qui elle-même envoie un message SIP *BYE* secondaire à tous les participants de la conférence (figure 7.8).

Notre passerelle a besoin des champs "*To*" et "*From*" du message. Ces deux valeurs seront utilisées par l'agent SIP pour identifier la conversation à terminer auprès de l'agent RTP.

A ce moment, l'agent RTP va stopper la conversation, envoyer le message dans une

requête HTTP à l'agent HTTP qui va l'envoyer sous forme de MMS toujours au format MM7.

7.2.3 Critique

Bien que théoriquement réalisable, ce scénario présente toutefois quelques limites dans la pratique.

La limitation principale concerne l'aspect volumétrique qu'engendrerait une telle conversation :

- la capacité de stockage de l'agent RTP doit être considérable pour pouvoir gérer plusieurs enregistrements simultanés ;
- la taille maximale d'un message est $\leq$ 300 kb (cfr figure 7.9). Le codec utilisé pour encoder le son est appelé AMR (Adaptative Multi-Rate). Selon [9] le débit de ce codec varie entre 4,75 et 12,2 kb/sec. Ceci signifie que dans le cas le plus favorable, la durée des messages ne pourrait dépasser $\frac{300kb}{4,75kb/sec}$ soit 63 secondes. Pour chaque conférence durant plus de 63 secondes, il faudrait diviser la conversation en minimum deux MMS. Ce qui est n'est guère pratique pour l'utilisateur recevant les messges.

Ce scénario n'est donc pas envisageable à l'heure actuelle. Une amélioration des technologies est souhaitable si l'on veut le mettre en pratique. Pour ce faire, il faudrait agir dans deux directions :

1. Augmenter la capacité en charge d'un MMS.
2. Améliorer les algorithmes de compression de données.

Table 5.4 Message content classes in the core message content domain

	Class text	Class image basic	Class image rich	Class video basic	Class video rich
Text	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16
Still image	None	Baseline JPEG (JFIF)	Baseline JPEG (JFIF)	Baseline JPEG (JFIF)	Baseline JPEG (JFIF)
Bitmap image	None	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP
Speech[a]	None	AMR narrowband	AMR narrowband	AMR narrowband	AMR narrowband
(Music) Audio	None	None	None	None	None
Synthetic audio	None	None	SP-MIDI	SP-MIDI	SP-MIDI
Video[b]	None	None	None	H.263 with AMR-NB (.3GP)	H.263 with AMR-NB (.3GP)
Vector graphics	None	None	None	None	None
Personal Information Manager	None	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar
Scene description	MMS SMIL	MMS SMIL	MMS SMIL	MMS SMIL	MMS SMIL
Support for OMA DRM – forward-lock	No	No	Yes	Yes	Yes
Message size	≤30 kB	≤30 kB	≤100 kB	≤100 kB	≤300 kB
Max image resolution	Not applicable	160 × 120	640 × 480	640 × 480	640 × 480

[a] AMR narrowband is the recommended speech codec for MMS clients conforming to 3GPP requirements. For MMS clients conforming to Third Generation Partnership Project 2 (3GPP2) requirements, OMA recommends the use of the 13-K speech codec instead.

[b] An H.263 video clip with an optional AMR audio track is transported in a 3GP file as defined by the 3GPP. Alternatively, 3GPP2 compliant MMS clients use an alternative file format known as 3GP2.

FIG. 7.9 – Tableau des caractéristiques d'un MMS (tiré de [4, page 84]).

Chapitre 8

Réponse d'un message en mode MULTIGROUP

Nous allons maintenant évoquer la possibilité pour un utilisateur non-IMS d'enregistrer un message vocal avec son téléphone mobile et de l'envoyer sous la forme d'un MMS à un utilisateur IMS. Cet utilisateur IMS recevra le message par l'intermédiaire de son application Push To Talk.

Nous partirons de l'hypothèse que l'utilisateur non-IMS envoie un MMS en réponse à un message reçu au préalable en provenance d'un utilisateur IMS. Nous prenons ce scénario dans un but de continuité par rapport au chapitre 7, qui évoquait l'envoi d'un message Push To Talk sous la forme d'un MMS de la part d'un utilisateur IMS. Toutefois, l'utilisateur non-IMS peut envoyer ce message de lui-même sans avoir reçu de message initial.

Le MMS envoyé par l'utilisateur non-IMS sera reçu par la PoC/MMS Gateway qui jouera le rôle d'un utilisateur IMS. Elle enverra uniquement des messages en mode MULTIGROUP, et ce, pour des raisons de latence que nous avons explicitées au chapitre 6.

Dans ce scénario, l'utilisateur non-IMS doit envoyer un MMS à destination d'une SIP URI et non pas d'un numéro de téléphone. Sans adaptation de l'environnement à ce genre de situation, aucun message ne pourra aboutir à destination. Nous montrerons comment il est possible de remédier à ce problème.

Notons enfin que la manipulation de SIP URI peut paraître à priori fastidieuse pour les utilisateurs et ce au même titre que les numéros de téléphone. Rarement avec notre mobile nous manipulons directement des numéros de téléphone. Nous nous référons le plus souvent au répertoire de notre téléphone qui associe un nom à un numéro. Il en ira de même pour les SIP URI.

8.1 Use case

Pré Condition :

- l'utilisateur non-IMS possède un mobile supportant le MMS et offrant la possibilité de dictaphone ;
- l'utilisateur non-IMS est connecté au système Push To Talk ;
- l'utilisateur non-IMS a reçu un MMS en provenance d'un utilisateur IMS.

Post Condition :

- l'utilisateur IMS reçoit un message Push To Talk en provenance de l'utilisateur non-IMS en mode MULTIGROUP.

Scénario :

Utilisateur	**PoC/MMS Gateway**
1. L'utilisateur non-IMS enregistre son message vocal et l'envoie dans un MMS.	
	2. La PoC/MMS Gateway reçoit le MMS et le transforme en message Push To Talk. 3. Elle envoie le message au destinataire.

Les pré-conditions supposent que l'utilisateur non-IMS dispose d'un téléphone suffisamment évolué pour pouvoir enregistrer sa voix et encapsuler le fichier son de l'enregistrement dans un MMS.

Reprécisons également que la pré-condition stipulant que l'utilisateur non-IMS a reçu au préalable un MMS en provenance d'un utilisateur IMS est là dans un unique but de continuité par rapport au chapitre précédent.

8.2 Diagrammes de séquence

Comme nous l'avons vu au chapitre 2, l'envoi d'un message Push To Talk est précédé de la réservation d'un jeton et suivi du relâchement de celui-ci (cfr les figures 2.5 et 2.6). Une requête SIP *INVITE* est envoyée au préalable, au moment de l'enregistrement (cfr chapitre 6).

Une fois l'invitation effectuée, les messages RTP/RTCP peuvent être envoyés. Ce rôle est dédié à notre agent RTP. Ce qui nous intéresse particulièrement dans ce chapitre, ce sont les échanges de messages qui précèdent les messages RTP/RTCP. Avant d'atteindre notre PoC/MMS Gateway, le MMS de notre utilisateur non-IMS passera par le MMSC de l'opérateur de téléphonie mobile qui le déviera vers notre passerelle.

Comme nous l'avons dit plus haut dans ce chapitre, le principal problème réside dans le fait que le MMS envoyé par l'utilisateur non-IMS contient comme numéro de téléphone une SIP URI. Le MMSC de l'opérateur qui a comme but, entre autres, de dévier les messages vers les bonnes personnes ne connaîtra pas ce numéro de téléphone et renverra un message d'erreur à l'émetteur du message. Il faut trouver un moyen de contourner ce problème.

Le MMSC développé par *Alcatel* est également basé sur la technologie de la Proxy Platform. Il s'agit d'un agent HTTP constitué de plusieurs proxylets sur la chaîne de requête et de réponse. Une solution serait d'ajouter une proxylet dans le flux de requête qui décompilerait le message et vérifierait le header "*From*". Lorsque celui-ci contient une SIP URI, la proxylet dévierait le message vers notre PoC/MMS Gateway.

Cette façon de faire ne fonctionne que si l'opérateur possède un MMSC produit par *Alcatel*. Il serait intéressant de généraliser l'idée afin qu'elle soit indépendante du MMSC.

Pour ce faire, il faudrait placer un agent HTTP contenant la même proxylet que celle décrite plus haut. Cet agent serait mis en amont du MMSC et intercepterait tous les messages afin de les dévier, si nécessaire, vers la PoC/MMS Gateway.

Après avoir montré une solution permettant de résoudre le problème lié au routage du message, nous allons étudier de plus près ce qui se passe au sein de la PoC/MMS Gateway une fois que celle-ci reçoit le MMS en provenance de l'utilisateur non-IMS.

La figure 8.1 nous montre le diagramme de séquence illustrant les interactions entre les différents composants de la PoC/MMS Gateway.

Lorsqu'une application cliente envoie un message Push To Talk, chaque message est entouré de deux fois deux messages. Un message SIP *INFO* avec comme sujet la valeur "*reserve*" et un message RTCP *SOS*. Une fois ces deux messages envoyés, le système Push To Talk attend le message vocal. Enfin, lorsque le message vocal est terminé, un message RTCP *EOS* et un message SIP avec comme sujet la valeur "*free*" sont envoyés pour annoncer la fin du message au système Push To Talk.

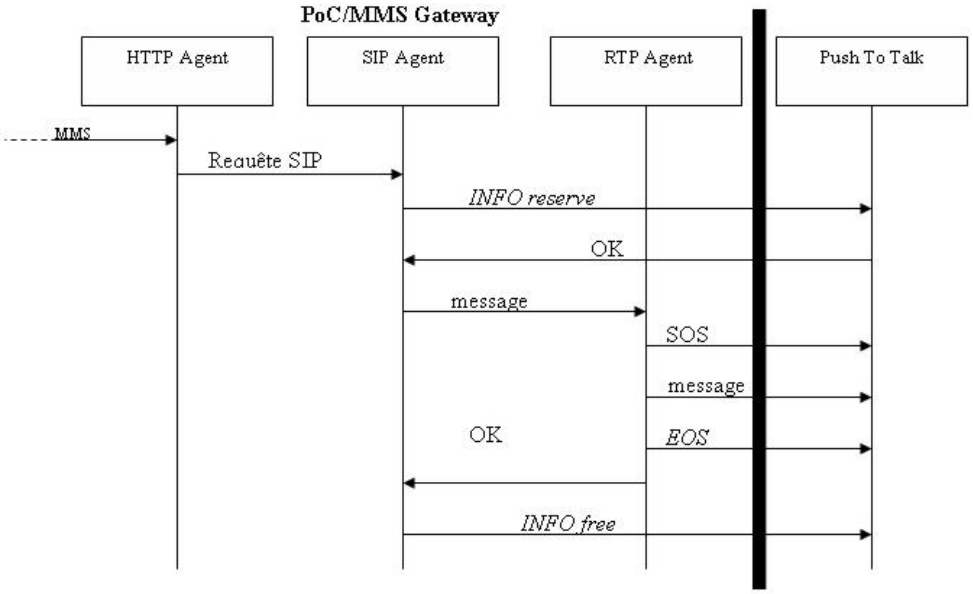


FIG. 8.1 – Interactions au sein de la PoC/MMS Gateway.

Notre passerelle jouera le rôle de l'application cliente et devra, par conséquent, effectuer toutes ces interactions.

A ce moment, nous avons besoin de trois choses :

1. La SIP URI de l'émetteur.
2. La SIP URI du récepteur.
3. Le message vocal de l'émetteur.

Ces informations seront respectivement présentes dans les headers "*From*","*To*" et dans le corps de la requête envoyée par le client SIP présent dans l'agent HTTP.

Il est important de préciser que comme nous sommes en mode MULTIGROUP, il y a création implicite d'une conférence éphémère au moment de la réservation du jeton. Par conséquent, la requête SIP *INFO* devra contenir la SIP URI du correspondant.

L'agent SIP se charge d'envoyer ce message SIP *INFO* avec comme sujet la valeur "*reserve*" au système Push To Talk. Une fois la réponse obtenue, l'agent SIP envoie le message à l'agent RTP. Celui-ci envoie tout d'abord le message *SOS* au système Push To Talk, suivi du message de l'utilisateur non-IMS et du message *EOS*. Il envoie enfin la réponse à l'agent SIP. Sur base de cette réponse, l'agent SIP peut finalement envoyer la requête *INFO* ayant comme sujet la valeur "*free*" au système Push To Talk pour lui signaler que le message est terminé.

Chapitre 9

Gestion de la présence

9.1 Le serveur de présence

Le serveur de présence est un élément clé dans une infrastructure IMS. Il a pour but d'agréger l'information de présence de différents systèmes et d'offrir une interface standard d'accès à cette information de présence.

Lorsqu'un utilisateur désire connaître l'information de présence d'un autre utilisateur, il doit y souscrire[1]. Suivant les règles d'autorisation de l'UAS, l'UAC pourra voir tout ou partie de l'information de présence publiée par celui-ci. Il existe en effet différents types d'information de présence :

- "*en-ligne/hors-ligne*" ;
- type de terminal ;
- type de connexion ;
- information sur le statut : affiche un message personnel ;
- localisation : "*au bureau*", "*à la maison*", ... ;
- ...

Les règles d'autorisation spécifient quelle information de présence peut être montrée à un UAC particulier. Elles sont traitées par l'intermédiaire de documents XML à travers le protocole XCAP.

Exemple : *Lorsque les gens sont en vacances, ils apprécient généralement oublier complètement leur travail et ne pas être dérangés par leurs collègues. Grâce aux règles d'autorisation, il leur est possible de cacher leur information de présence à leurs collègues.*

Le serveur de présence permet également de notifier, c'est-à-dire de diffuser l'information de présence selon les règles d'autorisation définies par un utilisateur à toutes les personnes ayant souscrit à son information de présence.

[1] Nous utiliserons les même notations que celles utilisées au chapitre 1. Nous appellerons UAC, l'utilisateur souscrivant à l'information de présence et UAS l'utilisateur souscrit.

Comme nous venons de le montrer, le serveur de présence est un outil qui permet de diffuser l'information de présence à toutes les personnes qui y ont souscrit. Il est vraisemblable que des utilisateurs seront amenés à souscrire à l'information de présence de plusieurs personnes. Il est donc intéressant de leur fournir une vue globale sur l'information de présence de tous leurs contacts. Cette vue globale se traduit par une liste de contacts. La gestion de cette liste se fait par le RLS. Il s'agit d'une couche qui se place au dessus du serveur de présence et qui permet à un utilisateur de gérer tous ses contacts.

Après avoir donné un aperçu succinct de la présence dans les IMS, nous allons analyser les scénarii qui nous seront utiles pour la conception de notre PoC/MMS Gateway. Nous montrerons comment un utilisateur publie son information de présence et la rend ainsi disponible à tous les utilisateurs qui y ont souscrit. Nous montrerons également comment un UAC souscrit à l'information de présence d'un UAS.

9.1.1 Publication de l'information de présence

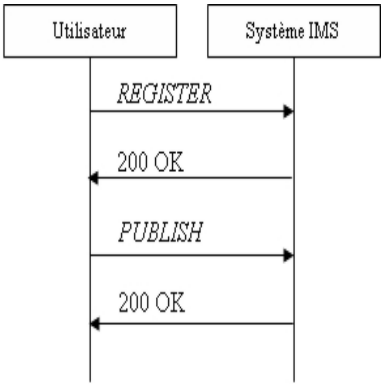


FIG. 9.1 – Publication de l'information de présence.

Comme le précise [5, page 311], la requête SIP *REGISTER* est suffisante pour publier une information de présence de base, c'est-à-dire "*en-ligne/hors-ligne*". Cependant, cette méthode n'est pas sémantiquement appropriée pour publier de l'information de présence plus complexe (donner le type de connexion, le type de terminal, la localisation, ...). Pour rappel, la requête SIP *REGISTER* est utilisée pour associer une adresse IP à une SIP URI. Par conséquent, un autre mécanisme est nécessaire. Celui-ci doit permettre de transférer des documents PIDF/RPID[2]. Comme nous l'avons vu au chapitre 1, la requête SIP *PUBLISH* est une requête générique permettant de publier l'état d'un événement. Cette requête se prête parfaitement à la publication de l'information de présence.

Dès lors, pour publier son information de présence, un utilisateur IMS envoie dans un premier temps une requête SIP *REGISTER* permettant de publier l'information de

[2]Il s'agit de protocoles utilisés pour structurer l'information de présence. Ces protocoles sont des applications du méta-langage XML. Nous n'entrerons pas dans le détail de la structure de ces documents, le lecteur pourra consulter [5, Ch 16] pour plus d'informations.

présence de base. Ensuite, il envoie une requête SIP *PUBLISH* permettant de publier une information de présence plus détaillée, celle-ci se trouvant dans un document PIDF/RPID (Figure 9.1).

Cette publication de présence est typiquement effectuée au démarrage de l'application cliente.

9.1.2 Souscription à l'information de présence

Le scénario est un peu plus complexe.

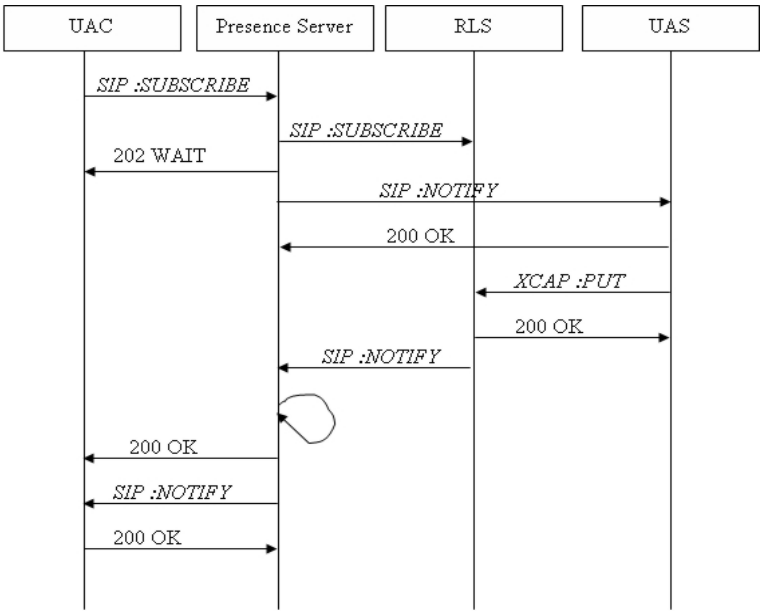


FIG. 9.2 – Souscription à la présence d'un utilisateur avant modification des règles d'autorisation.

Souscrire à la présence d'un utilisateur équivaut à ajouter une personne à sa liste de contacts, c'est pourquoi le RLS joue un rôle important dans ce scénario.

1. L'application cliente de l'utilisateur envoie une requête SIP *SUBSCRIBE* au serveur de présence. Ce message contient la SIP URI de l'UAS.
2. Le serveur de présence envoie une requête SIP *SUBSCRIBE* au RLS pour le prévenir qu'un changement va se produire.
3. Le serveur de présence vérifie les règles d'autorisation de l'UAS et constate que l'UAC n'en fait pas partie. Il va donc falloir demander à l'UAS s'il accepte de diffuser son information à l'UAC.
4. Le serveur de présence renvoie un code 202 WAIT à l'application cliente de l'UAC.
5. Le serveur de présence envoie une requête SIP *NOTIFY* à l'UAS. Concrètement, une fenêtre va apparaître sur l'interface du terminal de l'UAS, lui demandant s'il accepte

de diffuser son information de présence à l'UAC. L'UAS devra accepter ou refuser. Supposons qu'il accepte.

6. L'application cliente de l'UAS renvoie une réponse 200 OK au serveur de présence.
7. Elle envoie également une requête XCAP *PUT* au RLS, pour lui signaler l'ajout d'un utilisateur dans la liste de contacts.
8. Le RLS répond par un 200 OK.
9. Il envoie également une requête *NOTIFY* au serveur de présence.
10. Le serveur de présence recalcule les règles d'autorisation de l'UAS.
11. Il répond par un 200 OK à l'application cliente de l'UAC.
12. Enfin, le serveur de présence envoie une requête SIP *NOTIFY* à l'UAC pour lui donner l'information de présence de l'UAS.
13. L'application cliente de l'UAC répond par un 200 OK .

L'UAC fait maintenant partie des règles d'autorisation de l'UAS. Désormais, lorsqu'il se connectera, la procédure sera plus légère, comme nous le montre le diagramme de séquence de la figure 9.3 :

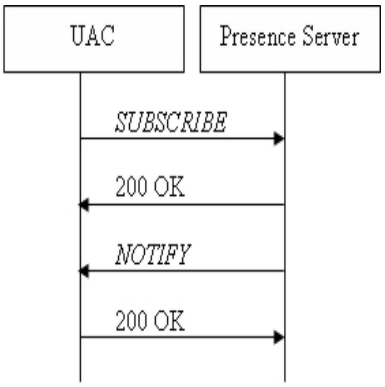


FIG. 9.3 – Souscription à la présence d'un utilisateur après modification des règles d'autorisation.

Nous avons montré le cas où l'UAS acceptait la souscription. Ceci dit, le scénario alternatif existe et il est illustré à la figure 9.4

Tout se déroule comme dans le scénario alternatif jusqu'au point 6. Le serveur de présence ne renvoie pas un code 200, mais un code d'erreur. Ce code d'erreur est également renvoyé à l'application cliente de l'UAC souscrivant.

9.2 Analyse de la présence

Dans cette section, nous allons étudier la gestion de la présence pour un utilisateur non-IMS.

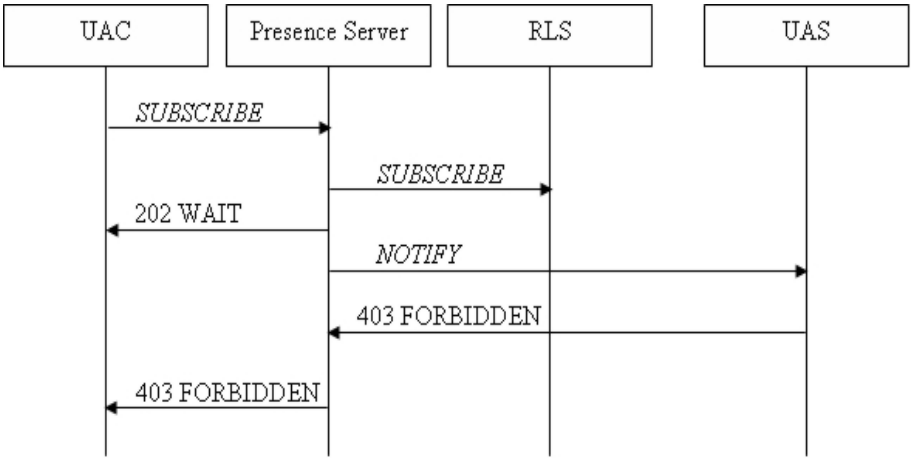


FIG. 9.4 – Refus de dévoiler son information de présence.

Nous proposerons une solution lui permettant d'une part de souscrire à l'information de présence d'un utilisateur IMS, d'autre part d'accepter de diffuser son information de présence à tout utilisateur désirant y souscrire.

Il est utile de permettre à un utilisateur non-IMS de souscrire à la présence d'utilisateurs IMS. En effet, s'il désire envoyer un message, il doit savoir si le destinataire est présent ou pas.

Une possibilité serait d'effectuer un "*appel en absence*". Cependant, un utilisateur IMS peut être connecté au réseau sans pour autant avoir démarré son application Push To Talk. De plus, si l'utilisateur IMS n'est pas connecté au réseau, l'utilisateur non-IMS sera facturé d'un appel inutile.

Une autre possibilité serait l'envoi d'un SMS Premium. En réponse à ce SMS, l'utilisateur non-IMS recevrait l'information de présence de l'utilisateur IMS. L'utilisateur non-IMS devrait procéder de la sorte dès qu'il désire connaître une information de présence. Ceci est beaucoup moins pratique qu'une liste de contacts qui donne une vue d'ensemble de tous les contacts en temps réel. Nous n'émulerons pas une telle liste volontairement. Comme nous l'avons précisé au chapitre 6, il ne faut pas offrir toutes les possibilités de l'IMS à l'utilisateur non-IMS, car si c'est le cas, il n'aura pas le désir de passer à l'IMS. De plus, nous limiterons l'information de présence d'un utilisateur IMS au simple "*en-ligne/hors-ligne*". Avec ce système, nous lui mettons l'eau à la bouche en lui offrant une partie des possibilités de l'IMS.

Il faut cependant constater qu'une telle solution serait coûteuse pour l'utilisateur non-IMS, vu le prix d'un SMS Premium. Dès lors, nous pensons qu'un opérateur devrait offrir à ses utilisateurs des forfaits leur permettant d'envoyer autant de SMS (liés à ce service) qu'ils le désirent.

Nous analyserons les trois situations suivantes :

1. Un utilisateur non-IMS désire souscrire à la présence d'un autre utilisateur. Pour

ce faire, il enverra un SMS Premium. Il recevra en réponse un SMS lui donnant l'information de présence de l'utilisateur.

2. Un utilisateur IMS désire souscrire à la présence d'un utilisateur non-IMS. L'utilisateur non-IMS reçoit un SMS lui demandant s'il accepte de publier son information de présence à cet utilisateur. La réponse à ce SMS indiquera si l'utilisateur non-IMS accepte ou non de publier sa présence.

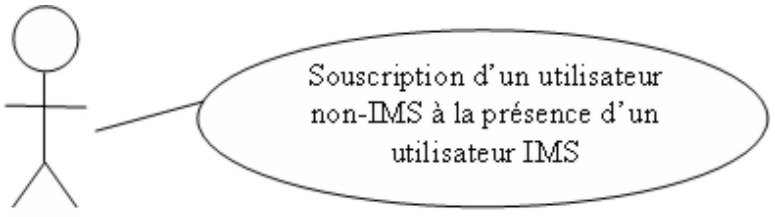
3. L'utilisateur non-IMS publie son information de présence. Ceci se fera implicitement au moment de l'enregistrement au système Push To Talk.

Pour chacune de ces situations, nous définirons un use case formel et analyserons l'interaction entre les différents éléments à l'aide de diagrammes de séquence.

9.2.1 Souscription à la présence d'un utilisateur IMS

Le scénario que nous présentons ici se déroulera de façon différente lors de la première souscription à la présence de l'UAS. En effet, la première fois, l'UAS devra accepter ou non de publier sa présence à l'UAC. Par la suite, le système aura mis à jour les règles d'autorisation de l'UAS et la présence sera publiée automatiquement. Ce mécanisme sera transparent aux yeux de notre utilisateur non-IMS.

Use Case



Pré Condition :

– l'utilisateur non-IMS possède un mobile supportant le SMS.

Post Condition :

– l'utilisateur non-IMS connaît l'information de présence de l'UAS.

Scénario :

Utilisateur	PoC/MMS Gateway
1.L'utilisateur non-IMS envoie un SMS Premium dans le but de souscrire à l'information de présence d'un UAS.	
	2. La PoC/MMS Gateway reçoit une requête HTTP en provenance de la SMS/MMS Gateway. 3. Sur base de cette requête, elle souscrit à la présence de l'UAS auprès du serveur de présence. 4. Elle reçoit l'information de présence de l'UAS. 5. Elle envoie un MMS contenant l'information de présence de l'UAS.
6. L'utilisateur non-IMS reçoit un SMS lui fournissant l'information de présence de l'UAS.	

Si c'est la première fois que l'utilisateur non-IMS souscrit à la présence de l'UAS, il ne recevra de réponse qu'au moment où celui-ci se connectera (sauf s'il est déjà connecté).

Nous allons permettre à l'utilisateur non-IMS, comme nous l'avons fait lors de la phase d'enregistrement, de souscrire à la présence d'une personne par le biais d'un SMS. Il enverra par exemple le mot-clé **SUBSCRIBE** suivi de la SIP URI de l'utilisateur pour lequel il désire souscrire la présence à un numéro court tel que 3310. Pour permettre ce genre de scénario, nous allons à nouveau pouvoir profiter de la SMS/MMS Gateway qu'il faudra configurer en conséquence.

Diagrammes de séquence

Le diagramme de séquence décrivant la souscription à la présence d'un utilisateur est illustré à la figure 9.2. Nous analyserons les interactions qui se trouvent en amont de ce schéma.

Notons d'emblée que deux situations sont possibles suivant le fait que ce soit ou non la première fois que l'utilisateur non-IMS souscrit à l'information de présence de l'UAC.

Lors de la première souscription, la PoC/MMS Gateway recevra un code 202 WAIT avant de recevoir le code 200 (OK). Ensuite, elle recevra immédiatement le code 200.

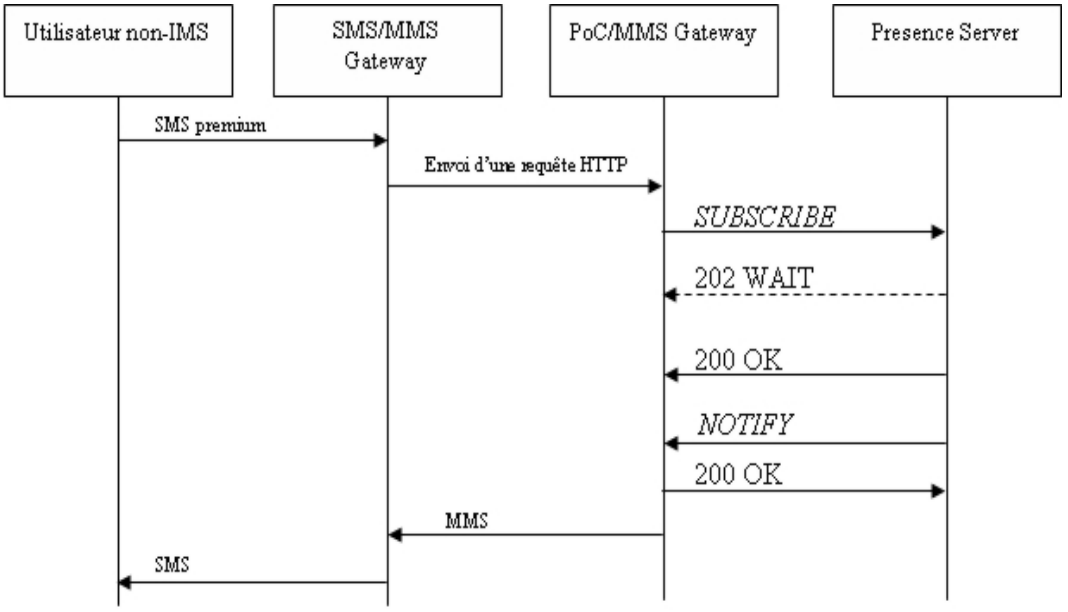


FIG. 9.5 – Souscription d'un utilisateur non-IMS.

Nous avons jusqu'ici raisonné sous l'hypothèse que l'UAS acceptait de diffuser son information de présence. Or celui-ci peut refuser. La figure 9.6 illustre ce cas.

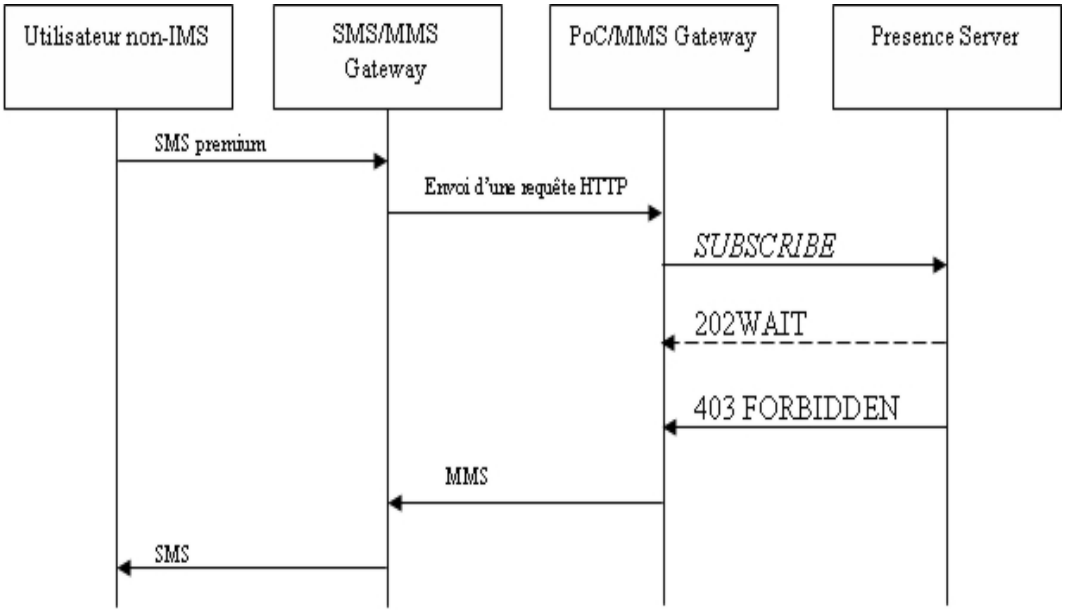


FIG. 9.6 – Refus de dévoiler son information de présence.

Notre passerelle recevra dans ce cas un code d'erreur 403 (précédé éventuellement d'un code 202). Elle n'attendra pas de *NOTIFY* du serveur de présence et renverra un message à l'utilisateur non-IMS lui indiquant que la personne n'est pas connectée.

Analysons maintenant de plus près ce qui se passe au sein de la PoC/MMS Gateway. Partons de celle-ci et posons-nous la question de savoir quelles informations elle requiert

pour souscrire à la présence d'un utilisateur.

Elle va devoir envoyer un message SIP *SUBSCRIBE* contenant deux informations, la SIP URI de l'UAC et la SIP URI de l'UAS. Comme pour le scénario d'enregistrement de l'utilisateur au système, l'utilisateur enverra un SMS Premium contenant un mot-clé accompagné de deux SIP URI, la sienne et celle de l'UAS.

Nous allons à nouveau pouvoir profiter des *mappings* de type "*expression régulière*" pour configurer notre SMS/MMS Gateway. Si nous prenons le mot-clé **SUBSCRIBE**, l'expression régulière doit être :

SUBSCRIBE *\s+(\S+) \s+(\S+)$*

Lorsque la SMS/MMS Gateway reçoit un tel SMS, elle envoie une requête HTTP en direction de la PoC/MMS Gateway. Nous utilisons une requête HTTP spécifique contenant les paramètres qui nous seront utiles :

- un paramètre contenant la SIP URI de l'UAC ;
- un paramètre contenant la SIP URI de l'UAS.

Les requêtes HTTP issues du *mapping* **SUBSCRIBE** pourraient avoir l'allure suivante :

`http://subscribe?sipurifrom=SipUri\&sipurito=SipUri`

La figure 9.7 nous montre les interactions au sein de notre PoC/MMS Gateway.

Notre agent HTTP envoie un message SIP à l'agent SIP contenant les deux SIP URI. L'agent SIP s'occupe ensuite de la souscription auprès du serveur de présence en envoyant la requête SIP *SUBSCRIBE*. En fonction de la situation, il recevra ou non un code 202 avant le code 200.

Lorsqu'il reçoit la requête SIP *NOTIFY* du serveur de présence, il peut renvoyer l'information de présence auprès de l'agent HTTP qui l'encapsule dans un MMS et l'envoie à la SMS/MMS Gateway. La SMS/MMS Gateway se charge alors d'envoyer l'information de présence sous la forme d'un SMS à l'utilisateur non-IMS.

Si l'UAS refuse de dévoiler son information de présence, l'agent SIP reçoit un code 403 (précédé ou non d'un code 202). Il peut à ce moment renvoyer directement l'information de présence [3] jusqu'à l'utilisateur final en passant par l'agent HTTP et la SMS/MMS Gateway.

9.2.2 Souscription à la présence d'un utilisateur non-IMS

Nous allons maintenant analyser la situation dans le sens inverse. L'UAC est un utilisateur IMS alors que l'UAS est un utilisateur non-IMS. Nous devons laisser la possibilité

[3] Qui dans ce cas est une information de non présence puisque l'UAS apparaîtra non connecté.

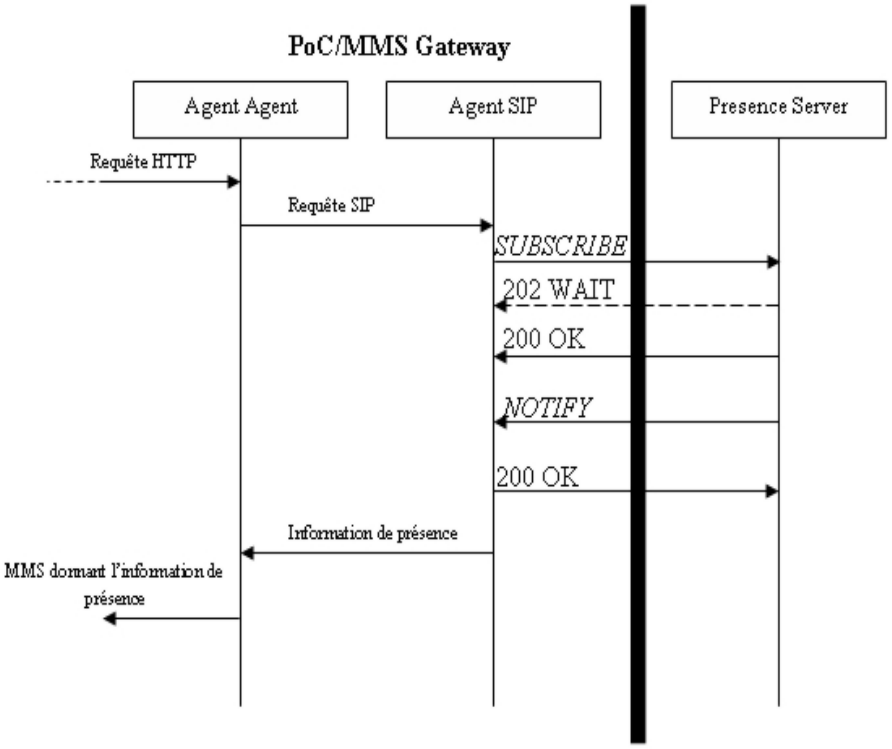
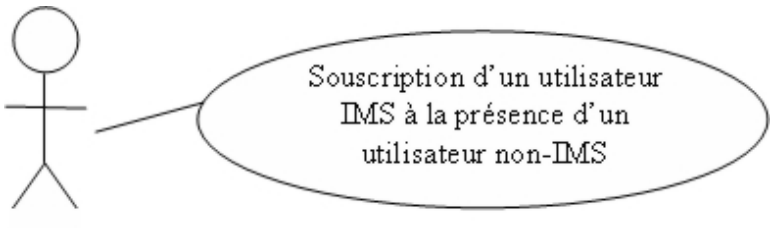


FIG. 9.7 – Interactions au sein de la PoC/MMS Gateway.

à l'utilisateur non-IMS d'accepter ou de refuser de diffuser son information de présence à quelconque UAC.

Lorsque l'utilisateur non-IMS se connecte au système, il publie son information de présence. S'il accepte de diffuser cette information à un utilisateur IMS, celui-ci sera tenu au courant, par l'interface de son logiciel client, lorsque l'utilisateur non-IMS se connectera.

Use Case



Pré Condition :

– l'utilisateur non-IMS possède un mobile supportant le SMS.

Post Condition :

– l'utilisateur a accepté ou refusé la souscription de l'utilisateur à son information de présence.

Scénario :

Utilisateur	PoC/MMS Gateway
1. L'utilisateur IMS souscrit à la présence d'un utilisateur non-IMS.	
	2. La PoC/MMS Gateway reçoit une requête SIP *NOTIFY* en provenance du serveur de présence. 3. Elle envoie un SMS à l'utilisateur final lui annonçant qu'un utilisateur a souscrit à son information de présence.
4. L'utilisateur non-IMS reçoit le SMS. 5. Il envoie SMS confirmant qu'il accepte ou refuse la souscription de l'utilisateur à son information de présence.	

Ce scénario ne se présentera qu'une seule fois pour chaque utilisateur souhaitant souscrire à l'information de présence d'un même utilisateur non-IMS. Une fois que celui-ci aura accepté ou refusé, c'est le serveur de présence qui répondra à la souscription grâce aux nouvelles règles d'autorisation.

Diagrammes de séquence

Dans cette section, nous nous baserons également sur la figure 9.2, qui contient le diagramme de séquence décrivant la souscription à la présence d'un utilisateur, mais nous l'analyserons dans le sens inverse. Nous n'analyserons pas les interactions qui se trouvent en amont mais en aval par rapport à ce schéma. Nous traiterons deux cas : celui où l'UAS accepte de diffuser son information de présence et celui où il refuse

L'UAS accepte de diffuser son information de présence La PoC/MMS Gateway reçoit une requête SIP *NOTIFY*. Sur réception de cette requête, elle va envoyer un message à l'utilisateur non-IMS pour l'avertir qu'une personne désire souscrire à son information de présence. Deux informations sont nécessaires :

1. Le numéro de téléphone de l'utilisateur non-IMS, afin de pouvoir envoyer le message au bon destinataire.
2. La SIP URI de l'UAC, afin de la renseigner à l'utilisateur non-IMS.

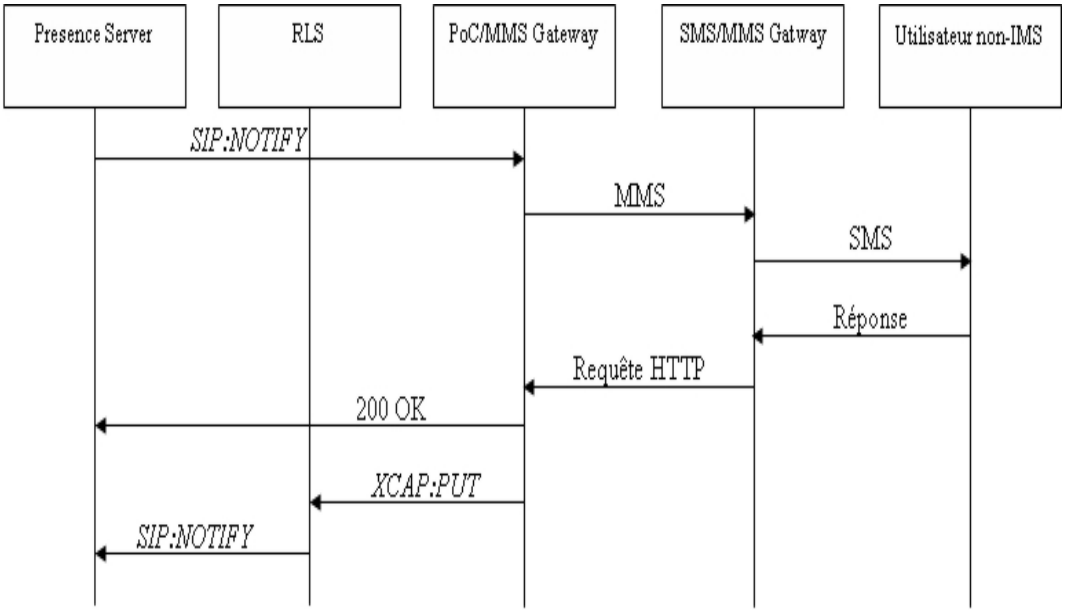


FIG. 9.8 – L'UAS accepte de diffuser son information de présence.

La PoC/MMS Gateway envoie un MMS en direction de la SMS/MMS Gateway qui se charge de le transformer en SMS pour l'envoyer à l'utilisateur final.

L'utilisateur doit répondre à ce SMS. Pour cela, il est nécessaire de définir un *mapping* de type "*expression régulière*" dans la SMS/MMS Gateway. Ce *mapping* prend deux arguments, la SIP URI de l'UAC et celle de l'UAS. Si nous prenons le mot-clé **ACCEPT**, l'expression régulière doit être :

ACCEPT *\s+(\S+) \s+(\S+)$*

La requête HTTP émanant de la SMS/MMS Gateway suite à la réception d'un tel SMS Premium aura l'allure suivante :

```
http://accept?sipurifrom=SipUri&sipurito=SipUri
```

Où :
- l'argument *sipurifrom* contiendra la SIP URI de l'UAS ;
- l'argument *sipurito* contiendra la SIP URI de l'UAC.

Sur base de cette requête HTTP, la PoC/MMS Gateway renvoie la réponse à la requête *NOTIFY* ainsi qu'un message XCAP *PUT*.

La figure 9.9 nous montre les interactions au sein de la PoC/MMS Gateway.

L'agent HTTP reçoit la requête émanant de la SMS/MMS Gateway. Sur base de cette requête, il envoie une requête SIP à l'agent SIP qui peut répondre à la requête SIP *NOTIFY* qui avait été envoyée par le serveur de présence.

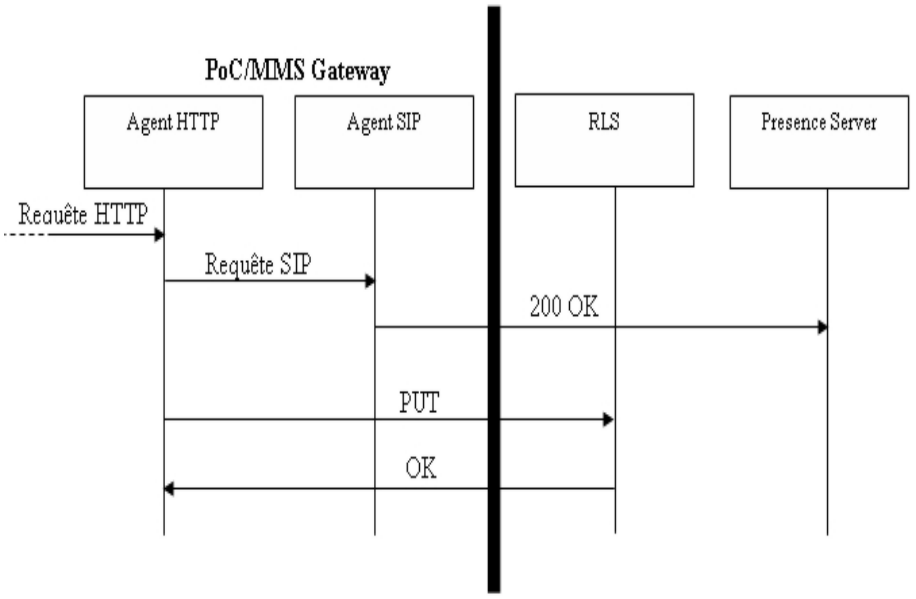


FIG. 9.9 – Interactions au sein de la PoC/MMS Gateway.

Il envoie également une requête XCAP au RLS. Celle-ci contient les deux SIP URI de la requête HTTP issue de la SMS/MMS Gateway. Cette requête a pour but d'ajouter l'utilisateur non-IMS à la liste de contacts de l'utilisateur IMS.

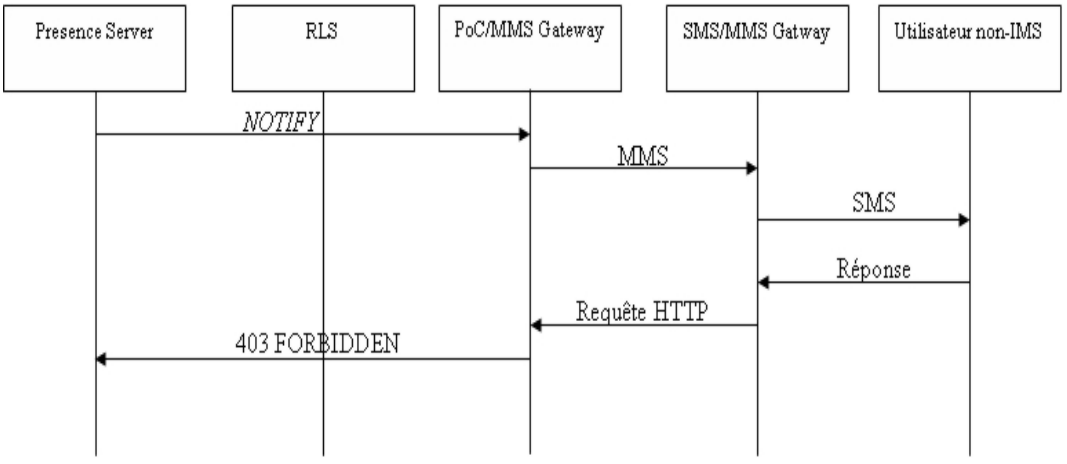


FIG. 9.10 – L'UAS refuse de diffuser son information de présence.

L'UAS refuse de diffuser son information de présence La situation est quelque peu différente. Pour offrir à l'UAS la possibilité de refuser l'information de présence, il faut définir un nouveau *mapping* de type expression régulière. Ce *mapping* prendra deux arguments, la SIP URI de l'UAC et celle de l'UAS. Si nous prenons le mot-clé **DENY**, nous devrons définir l'expression régulière suivante :

DENY *\s+(\S+) \s+(\S+)$*

La requête HTTP émanant de ce la SMS/MMS Gateway suite à la reception d'un tel SMS Premium aura l'allure suivante :

```
http://deny?sipurifrom=SipUri&sipurito=SipUri
```

Où :
- l'argument *sipurifrom* contiendra la SIP URI de l'UAS ;
- l'argument *sipurito* contiendra la SIP URI de l'UAC.

Sur base de cette requête HTTP, la PoC/MMS Gateway renvoie la réponse à la requête *NOTIFY* par un code d'erreur 403.

La figure 9.10 nous montre les interactions au sein de la PoC/MMS Gateway :

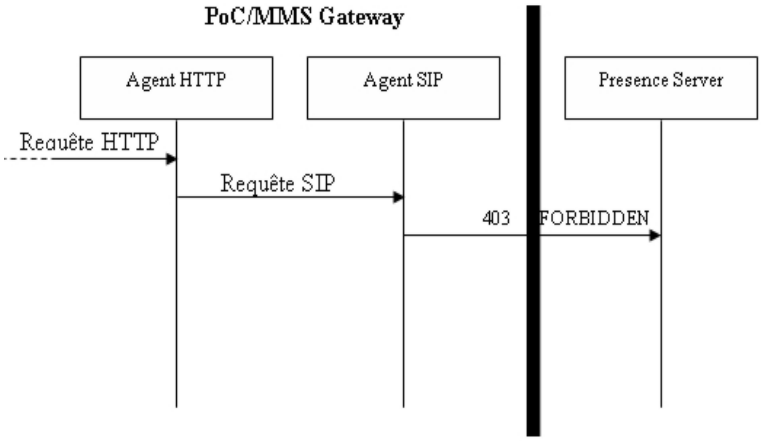


FIG. 9.11 – Interactions au sein de la PoC/MMS Gateway.

L'agent HTTP reçoit la requête émanant de la SMS/MMS Gateway. Sur base de cette requête, il envoie une requête SIP à l'agent SIP qui va répondre à la requête SIP *NOTIFY*, qui avait été envoyée par le serveur de présence, par un code d'erreur.

9.2.3 Diffusion de l'information de présence d'un utilisateur non-IMS

Comme nous l'avons précisé plus haut, c'est au moment du démarrage de l'application cliente que s'effectue la diffusion de l'information de présence. L'information de présence d'un utilisateur non-IMS se limitera volontairement à l'information "*en-ligne/hors-ligne*" [4]. Nous savons que pour diffuser une telle information, la méthode SIP *REGISTER* est suffisante. Dès lors, le système tel que nous l'avons conçu permet de diffuser l'information de présence de l'utilisateur non-IMS.

[4] De cette façon, l'utilisateur non-IMS n'aura pas les mêmes possibilités qu'un utilisateur IMS.

Conclusion

Nous avons initié le lecteur aux différents services de la téléphonie mobile d'aujourd'hui à savoir le SMS et le MMS, ainsi qu'aux services qui constitueront la téléphonie mobile de demain avec les IMS et plus précisément le Push To Talk. Pour ce faire, après avoir introduit les prérequis, nous avons présenté deux passerelles différentes.

La première passerelle, la SMS/MMS Gateway, fut implémentée au cours d'un stage réalisé au sein de la société *Nextenso S.A.* Cette solution permet à un opérateur de téléphonie mobile de gérer des services SMS et MMS Premium en partenariat avec un ou plusieurs fournisseurs de contenu. Des contraintes commerciales nous " imposaient " plus ou moins la façon dont la passerelle devait être implémentée. Dès lors, nous avons pris du recul et tenté de soulever les problèmes que pouvaient entraîner de telles contraintes.

A travers la deuxième passerelle, nous avons tenté de sensibiliser le lecteur aux problèmes auxquels sont confrontés les opérateurs de téléphonie mobile lorsqu'ils souhaitent changer les habitudes des consommateurs grâce à l'apport de nouvelles technologies. En l'occurrence, cette passerelle originale que nous avons baptisée PoC/MMS Gateway a pour but de permettre aux utilisateurs IMS de communiquer avec des utilisateurs non-IMS par le biais du MMS et du SMS.

Le domaine des IMS est tellement vaste que nous n'avons su en couvrir qu'une partie en nous limitant à l'application Push To Talk. De plus, nous n'avons pas fait le tour du problème. Ainsi quelques questions restent en suspens :

- entre le moment où l'utilisateur non-IMS reçoit la présence de l'utilisateur IMS et le moment où il envoie un message Push To Talk, plusieurs secondes peuvent s'écouler. Pendant ce temps, l'utilisateur IMS peut se déconnecter. Dans un tel cas nous pensons que deux pistes sont possibles :
  1. Envoyer un MMS à l'utilisateur IMS.
  2. Enregistrer le message dans la "*Voice Mail*" de l'utilisateur IMS ;
- il faut offrir à l'utilisateur non-IMS la possibilité de ne plus publier sa présence à une personne pour laquelle il avait au préalable accepté de la publier. Ceci pourrait également se faire par l'envoi d'un SMS Premium ;
- nous n'avons pas non plus abordé la partie gestion des utilisateurs non-IMS. Pour

bénéficier d'un tel service, les utilisateurs doivent y être autorisés. Cependant, cette partie relève de l'infrastructure IMS elle-même, comme nous l'avons montré dans à la figure 1.

Le but n'était pas de concevoir entièrement une passerelle afin que celle-ci soit implémentée et constitue un succès commercial. Nous voulions montrer en quoi cette passerelle serait faisable ou pas. Avec le recul, nous pensons d'ailleurs que, bien que réalisable, elle ne constituera pas un succès commercial, du moins dans l'état actuel des choses. Pour ce faire, il faudrait que du coté des terminaux des utilisateur non-IMS se développent des applications à l'interface attrayante qui permettraient, par exemple, aux utilisateurs de vérifier la présence d'une personne sans devoir envoyer un SMS. Les SMS seraient toujours envoyés derrière ces interfaces, mais cela serait transparent aux yeux des utilisateurs.

Nous pensons enfin que d'autres pistes existent pour permettre aux utilisateurs IMS de communiquer avec les utilisateurs non-IMS. Ainsi, une telle passerelle pourrait être évitée si le terminal envoyait le message Push To Talk sous la forme d'un MMS de manière transparente à l'utilisateur, lorsque le destinataire est un utilisateur non-IMS.

Bibliographie

[1] http ://www.dynamicsoft.com/index.php.

[2] Alcatel. Alcatel IMS Services IMS Enablers & Applications Description. 2004.

[3] Gwenaël Le Bodic. *mobile messaging technologies and services SMS,EMS and MMS.* John Wiley & Sons Ltd, 1st edition, november 2002.

[4] Gwenaël Le Bodic. *multimedia messaging service an engineering approach to MMS.* John Wiley & Sons Ltd, 1st edition, october 2003.

[5] Gonzalo Camarillo. *The 3G IP Multimedia Subsystems Merging The Internet And The Cellular Worlds.* John Wiley & Sons Ltd, 1st edition, december 2004.

[6] James F.Kurose and Keith W.Ross. *Computer Networking A Top-Down Approach Featuring the Internet.* Addison Wesley, 2nd edition, september 2003.

[7] J.Rosenberg, H.Schulzrinne, G.Carmarillo, A.Johnston, J.Peterson, R.Sparks, M.Handley, and E.Schooler. RFC 3261 - SIP : Session Initiation Protocol. Technical report, IETF, July 2002. http ://www.rfc-editor.org/rfc/rfc3261.txt.

[8] J.Rosenberg, H.Schulzrinne, U.Columbia, M.Handley, and E.Schooler. RFC 2543 - SIP : Session Initiation Protocol. Technical report, IETF, March 1999. http ://www.rfc-editor.org/rfc/rfc3261.txt.

[9] J.Sjoberg, M.Westerlund(Ericsson), A.Lakaniemi(Nokia), and Q.Xie(Motorola). RFC 3267 - Real-Time Transport Protocol (RTP) Playload Format and Files Storage Format for the Adaptative Multi-Rate (AMR) and Adaptative Multi-Rate WideBand (AMR-WB) Audio Codecs. Technical report, IETF, June 2002. http ://www.rfc-editor.org/rfc/rfc3267.txt.

[10] Renaud Marquet. Optimisation d'un logiciel de transmission de MMS. Master's thesis, Facultés Universitaires Notre-Dame de la Paix, Namur, 2004.

[11] Nextenso. Proxy Platform 2.0 Reference Guide. Juin 2003.

[12] Nextenso. A 5317 Push To Talk 1.1 Specification (v1.6-DRAFT). september 2004.

[13] Mathieu Van Overstraeten. Les "*SMS Premiums*" pèsent 50 millions. *La Libre Belgique*, 2003.

[14] Miikka Poikselkä, Georg Mayer, Hisham Khartabil, and Aki Niemi. *The IMS IP Multimedia Concepts and Services in the Mobile Domain.* John Wiley & Sons Ltd, 1st edition, april 2004.

[15] Henning Schulzrinne. Site de Henning Schulzrinne. [en ligne] disponible sur : http ://www.cs.columbia.edu/ hgs/sip (Dernière visite le 10 mai 2005).

[16] Laurent Schweizer. Sip & Mobility. Technical report, TCOM, 2001. http ://www.tcom.ch/Tcom/Projets/VoIP/VoIP_and_Mobility/-Travaux_de_diplomes/Rapport_Schweizer.pdf.

Annexes

Annexe A

Architectures des différents réseaux

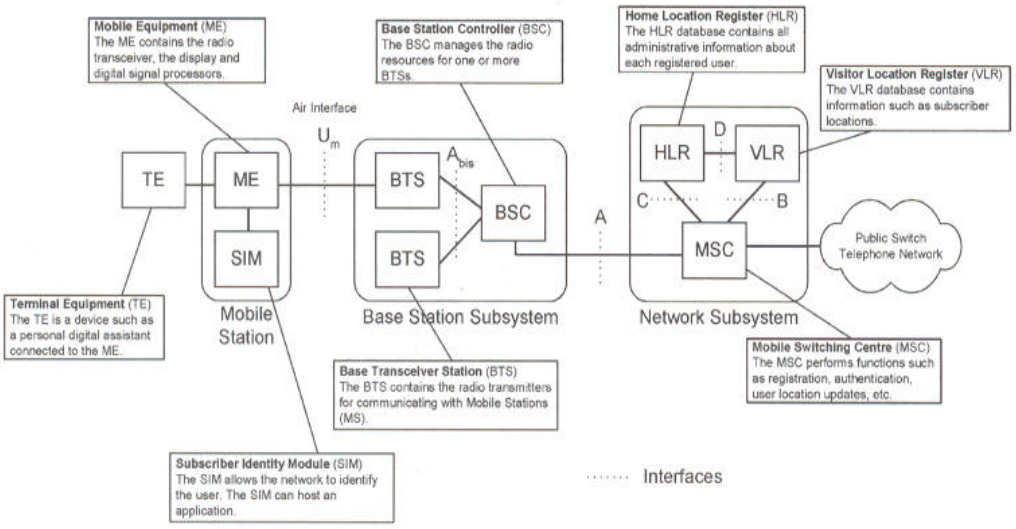


FIG. A.1 – Architecture d'un réseau GSM (tiré de [3, page 4])

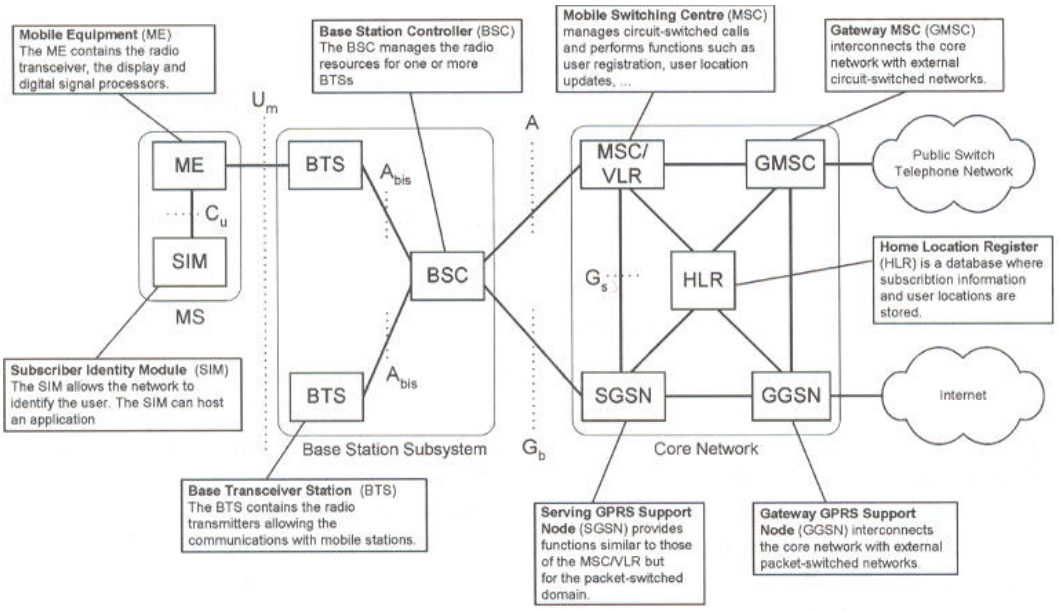


FIG. A.2 – Architecture d'un réseau GPRS (tiré de [3, page 8])

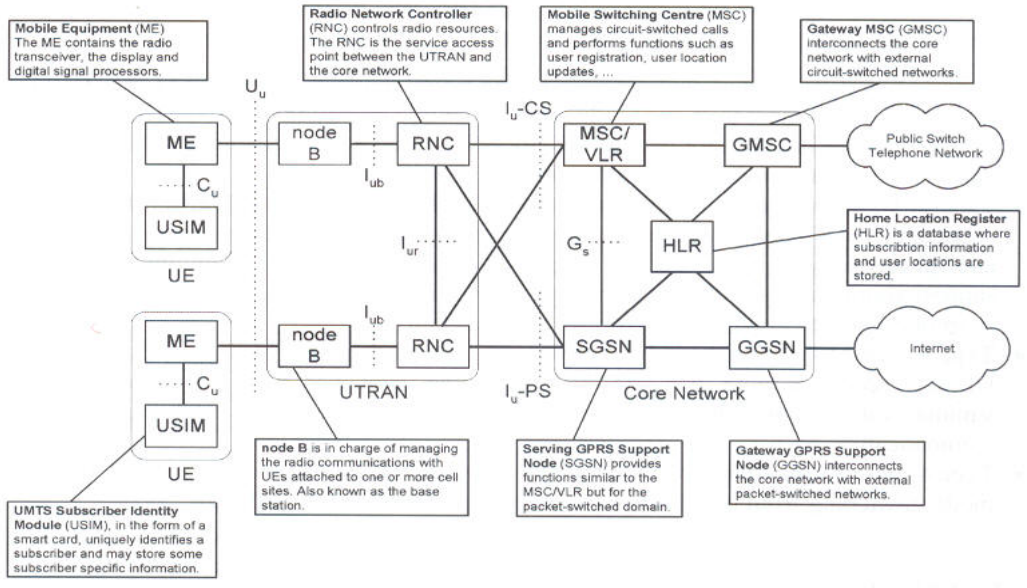


FIG. A.3 – Architecture d'un réseau UMTS (tiré de [3, page 11])

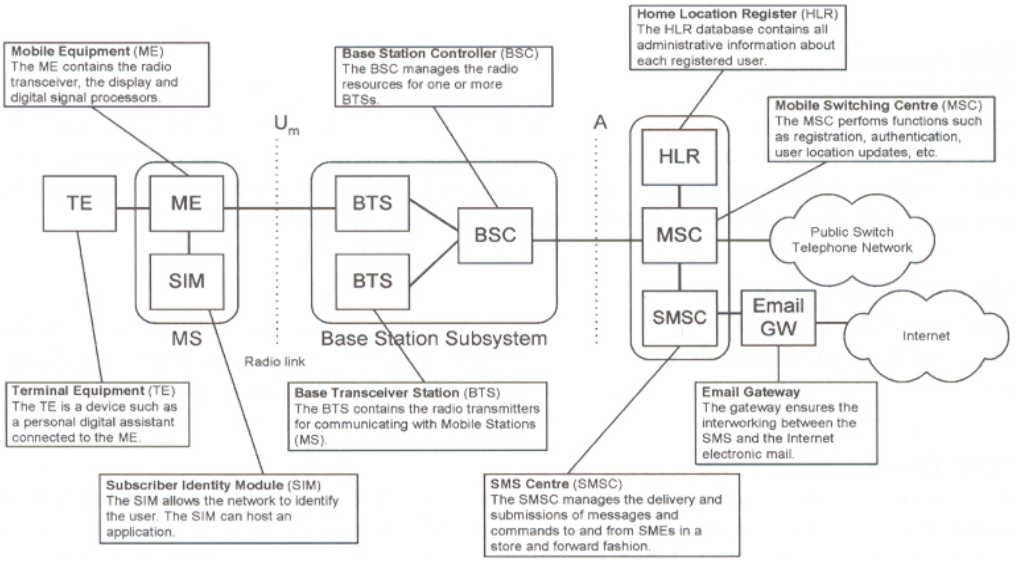


FIG. A.4 – Architecture SMS (tiré de [3, page 40])

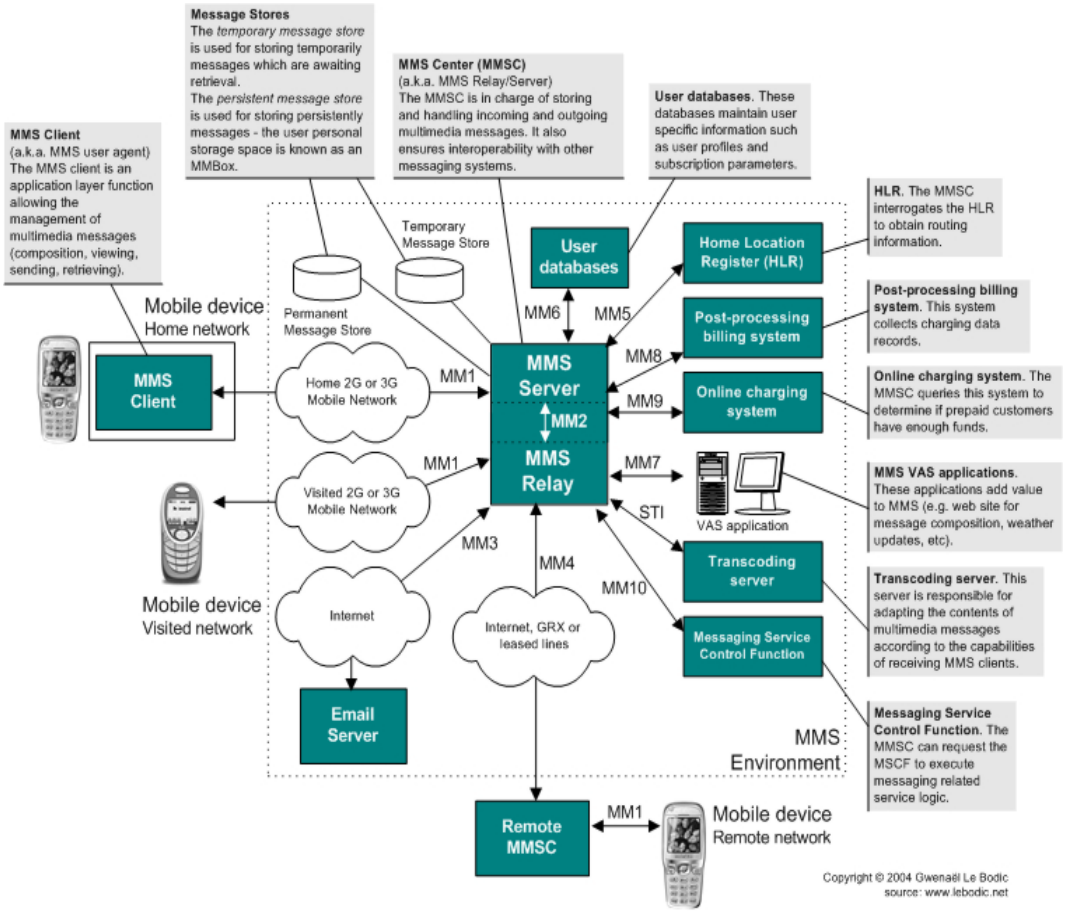


FIG. A.5 – Architecture MMS (tiré de [4, page 36])

Annexe B

Réponses SIP

Information, 1xx

- 100 en cours d'essai (trying) ;
- 180 sonne (ringing) ;
- 181 en cours de transfert ;
- 183 en cours d'exécution.

Succès, 2xx

- 200 OK ;

Redirection 3xx

- 300 choix multiples ;
- 301 déplacement permanent ;
- 302 déplacement temporaire.

Erreur du client 4xx

- 400 mauvaise demande ;
- 401 non autorisé ;
- 403 interdit ;
- 404 pas trouvé ;
- 407 autorisation du proxy demandé ;
- 408 temps pour la demande écoulé (request timeout) ;
- 420 mauvaise extension ;
- 480 momentanément pas disponible ;
- 481 call leg doesn't exist ;
- 482 détection de boucle ;

- 483 trop de saut (transfert, déviation) ;
- 484 adresse incomplète ;
- 485 ambigus ;
- 486 occupé ;
- 487 demande annulée ;
- 488 pas acceptable.

Erreur du serveur 5xx

- 500 Erreur interne du serveur ;
- 501 Pas implémenté ;
- 502 Mauvais gateway ;
- 503 Services non disponibles ;
- 504 Temps du gateway écoulé ;
- 505 Version pas supportée.

Erreur générale 6xx

- 600 occupé ;
- 601 décliné ;
- 604 n'existe pas ;
- 605 pas acceptable.